

AÑO II
Núm. 13
Mayo 1986
300 Ptas.

Test: Yamaha CX5M y CX5M II

SVI-318/328: Análisis interno

Código máquina, Basic

Libros, Programas,...

VG-8235
LA II GENERACION EN MARCHA

DIRECTOR:
Juan Arencibia.
COORDINADOR EDITORIAL:
J. Ignacio Rey.
COLABORADORES:
Octavio López, Angel Zarazaga, Teresa Aranda, Ricardo García.
DISEÑO:
Benito Gil
Editada por:
PUBLINFORMATICA, S.A.
C/ Bravo Murillo, 377 - 5.º A
Tel.: 733 74 13
28020 Madrid.
Telex 48877 OPZXE
PRESIDENTE:
Fernando Bolin.
DIRECTOR EDITORIAL REVISTAS DE USUARIOS:
Juan Arencibia.
DIRECTOR DE VENTAS:
Antonio González.
JEFE DE PRODUCCIÓN:
Miguel Onieva.
SERVICIO AL CLIENTE:
Julia González.
Tel.: 733 79 69
DIRECCION, REDACCION Y ADMINISTRACION:
C/ Bravo Murillo, 377 - 5.º A.
Tel.: 733 74 13
28020 Madrid.
PUBLICIDAD EN MADRID:
Emilio García.
PUBLICIDAD EN BARCELONA:
Lidia Cendros.
C/ Pelayo, 12.
Tel.: (93) 301 47 00 Ext. 27-28.
08001 Barcelona.
Depósito Legal: M. 16.755-1985
Imprime: G. Velasco, S. A.
Distribuye:
S.G.E.L. Avda. Valdelaparra, s/n.
Alcobendas (Madrid).
DISTRIBUIDORES:
VENEZUELA: SIPAM, S.A. Avda. República Dominicana, 541
ARGENTINA: DISTRIBUIDORA INTERCONTINENTAL BUENOS AIRES.

El P.V.P. para Ceuta, Melilla y Canarias, incluido servicio aéreo será de 300 ptas. sin I.V.A.

SUSCRIPCIONES
Rogamos dirija toda la correspondencia relacionada con suscripciones a
MSX
EDISA Tel. 415 97 12
C/López de Hoyos, 141 5.º
28002 MADRID
(Para todos los pagos reseñar solamente MSX)
Para la compra de ejemplares atrasados diríjanse a la propia editorial
MSX
C/Bravo Murillo, 377 5.º A
Tel. 733 74 13 28020 MADRID

Si deseas colaborar en MSX remite tus artículos o programas a Bravo Murillo 377, 5.º A 28020 Madrid. Los programas deberán estar grabados en cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada colaboración aisladamente, estudiando su complejidad y calidad



Estamos en un momento cumbre del estándar MSX. Junto con la creciente cantidad de software que está apareciendo, llega la 2.ª generación de la mano de Philips.

Es, sin lugar a dudas, una etapa importante la que se inicia y más teniendo en cuenta la fuerte competencia que existe actualmente. Ordenadores de 128K que ofrecen unas buenas prestaciones a buen precio y la aparición de nuevos equipos, ha hecho del mercado español un auténtico coto de caza para los fabricantes. Ya no basta con aparentar tener un buen aparato, hay que demostrarlo y comparar. No sin razón se dice con frecuencia que las comparaciones son odiosas ¿por qué será?, sin embargo, la mayoría de los usuarios, por falta de tiempo o por cualquier otro motivo compran mirando la cartera antes que las prestaciones que ofrece determinado ordenador. Esta es, por desgracia, una práctica frecuente y a la que está sujeta la gran mayoría.

Este mes, y rompiendo los moldes, ofrecemos como primicia el VG-8235, el primer ordenador de la 2.ª generación. El lector se preguntará por qué motivo, un ordenador de una casa concreta aparece como tema principal y no en la sección de TEST. La razón es muy sencilla. Después de varios meses de indagar, buscar y ver un ordenador de la 2.ª generación (en el Sonimag del año pasado), hemos tenido la gran oportunidad de disfrutar de un ordenador de estas características durante un tiempo y cómo no, teníamos que contárselo a alguien. ¿A quién mejor que nuestros lectores? Al fin y al cabo, ésta es una revista en la que todos tenemos algo que ver y como tal debe servir, tanto a los lectores como a los fabricantes como vehículo para expresar ideas y exponer las críticas (ya sean buenas o malas, aunque las buenas no interesan). En suma, nuestra obligación para con vosotros es informar de las importantes innovaciones que están en marcha, y qué menos que hacerlo con más detalle que en otras ocasiones.



SUMARIO





NOTICIAS

Informat '86, una cita con el futuro

Entre los días 13 y 17 de mayo, se celebrará en la Feria de Barcelona, una nueva edición del Salón de la Informática Informat '86. Este año serán 124 las empresas expositoras que representarán a 250 marcas, en una superficie de 8.343 metros cuadrados.

La confirmación de su creciente peso específico e influencia en el mercado español de informática y su mantenimiento como salón netamente dirigido a los visitantes profesionales del sector, coinstituyen los dos objetivos básicos a verificar por Informat '86, que se desarrollará simultáneamente con Expotrónica '86, el Salón de la Electrónica Profesional, Equipos y Componentes que organiza Feria de Barcelona. Informat pretende cubrir un vacío en cuanto a la existencia en nuestro país de ferias informáticas exclusivamente dedicadas a los profesionales. En él tienen cabida los importadores oficiales, y se ha excluido a los importadores paralelos aún cuando muchos de ellos son totalmente legales.

Problemas con algunas tarjetas inteligentes

Desde la comercialización en nuestro país de las tarjetas inteligentes, por parte de la empresa SERMA, se han detectado algunas con defectos. Puestos en contacto con el distribuidor oficial de estas tarjetas, aún no se sabe el motivo que ha ocasionado este error, por lo que cualquier usuario que halla tenido algún tipo de problema deberá ponerse en contacto con SERMA, C/ Bravo Murillo, 3-A. 28020 Madrid, y no remitirse a la tienda donde la adquirió.

Acuerdo entre ACE Software S.A. y KUMA, C.P. Soft y BUBLE BUS

La importante empresa de software ACE Software S.A. ha llegado a un acuerdo con las firmas anteriores para comercializar en España una serie de programas entre lo que hay dos dedicados a aplicaciones profesionales, como son Tratamiento de Textos y una Hoja de Cálculo. El resto de los programas son de entretenimiento y se titulan: Kubus, Buster Block y Zipper.

Idealogic lanza una serie de aventuras y misterio para ordenador

Idealogic anuncia el programa «Profesión: Detective», perteneciente a una serie de títulos de aventura y misterio en donde se produce un enigma y el usuario ha de resolverlo, buscando pistas, haciendo deducciones, obteniendo declaraciones, etc. El programa trabaja con un pequeño vocabulario en castellano y dispone de gráficos animados. El primer caso es el llamado: «El fantasma de Villa del Mar». Aquí el Detective dispone de un Ordenador Central en donde archivar los datos obtenidos, un Detectomóvil para desplazarse por la ciudad, una Detecticámara Fotográfica y multitud de distintos objetos y pantallas por las que desplazarse.

Con este programa se ha creado, a su vez, el concurso «Busca el misterio» para todos aquellos poseedores del programa que deberán contestar a las preguntas:

a) Nombre del culpable.
b) ¿Cómo lo hizo?
c) ¿Por qué?

Las bases de participación van adjuntas a cada uno de los programas y estarán disponibles en todos los distribuidores de esta empresa.

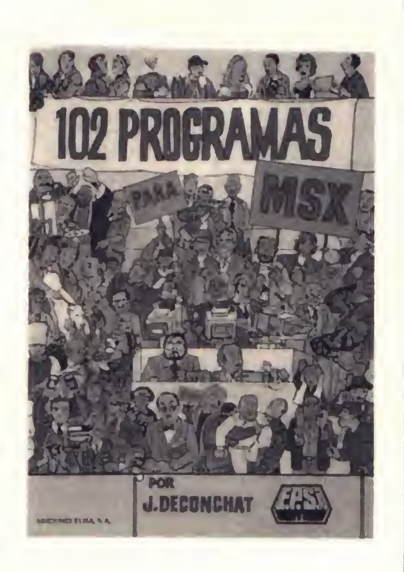
La fecha tope para enviar las soluciones será el próximo día 15 de julio, dándose amplia información de los ganadores.

Para obtener más información, dirigirse a Idealogic, S.A.
C/ Valencia, 85
08029 Barcelona

Aclaración

En el número anterior de MSX Magazine, un error de transcripción hizo que, en la sección de libros de la pág. 18 apareciera un comentario que no se corresponde con la cabecera y la foto que aparece en nuestra revista.

El comentario que aparece corresponde al libro: «102 Programas para MSX», del que es autor J. Deconchat, y no a «MSX Lenguaje Máquina» de Dullin-Strassenburg que fue lo que apareció a causa de un lamentable error.



Primera reunión de fabricantes de MSX

A principios del mes de Abril, se formalizó el primer acuerdo de colaboración entre las diferentes marcas que están presentes en nuestro mercado con equipos MSX. Participaron en la reunión celebrada en Barcelona las siguientes marcas:

CANON, DYNADATA, MITSUBISHI, PANASONIC, PHILIPS, PIONEER, SANYO, SONY y TOSHIBA.

Después de un análisis general de la situación actual del mercado de los ordenadores domésticos y de los MSX en particular, se establecieron las bases para la futura colaboración marcando de forma clara y precisa cuáles son los aspectos comunes a todas las marcas y los que deben promover de forma conjunta. Entre estos aspectos figuran de modo principal:

— Difusión del MSX como primer estándar.
— Apoyo conjunto a la red de ventas.
— Difusión del catálogo actual de programas, aunque muy numeroso (cerca de 400 títulos), muy desconocido debido a la dispersión de las marcas.
— Participación conjunta en ferias.
— Difusión de noticias y artículos referentes al estándar.

Los objetivos fijados deben dar unos primeros resultados a corto plazo, procurando que sean realistas al máximo, a fin de que la mencionada colaboración sea desde el primer momento una realidad.

VG 8235





La tan anunciada 2.ª Generación ya está en marcha y Philips ha sido la primera empresa en romper el hielo para bien de los usuarios del estándar. Sobre todo a los que esperábamos estos ordenadores, claramente superiores a los comercializados hasta ahora.

Desde hace ya varios meses, veníamos insistiendo en la aparición de una versión mejorada de los MSX actuales, denominados MSX 2, cuyas características vamos a explicar en esta ocasión con un auténtico 2.ª Genaración delante. El tiempo nos ha dado la razón y hemos podido comprobar como, poco a poco el estándar se ha abierto camino en el ámbito de los ordenadores personales. Ahora se inicia una nueva etapa, una auténtica lucha para dominar el mercado de los 128K, donde varios fabricantes han conseguido colocar sus productos en una importante posición y a un buen precio.

Sin embargo, los aparatos que existen actualmente, tienen unas cualidades sensiblemente mejoradas con respecto al modelo base, mientras que en el MSX 2, ésta mejora no es sólo parcial, sino total, en aquellos casos estamos ante el mismo perro pero con distinto collar, mientras que en la 2.ª Generación no sólo se ha cambiado de perro sino también de collar. Esto se debe a que estamos ante un ordenador con una clara orientación semiprofesional, con ventajas que hasta el momento no habíamos encontrado en otra máquina.

Philips ha dado un paso muy importante al lanzar este producto, que podríamos decir es un auténtico ordenador de la 2.ª Generación (descafeinada, puesto que para los amantes del vídeo y los forofos videoclips, habrá otra versión con una unidad de superposición, con el que se podrá realizar cualquier tipo de montaje de este tipo sobre un vídeo disco, algo improbable para el usuario medio). No habrá diferencia entre uno y otro modelo salvo el precio y la

...n
...eva
...eración

mencionada facilidad de superponer imágenes de vídeo con otra del televisor, un magnetoscopio o una cámara de vídeo. En realidad, es el mismo aparato sólo que preparado para esas personas que lo utilizan para algo más que programas, jugar, etc. Es una herramienta, que será de suma utilidad para aquellos que aprovechan las potentes cualidades que ofrecen un ordenador y un vídeo disco.

El modelo sobre el que vamos a basar este artículo es el VG-8235, una máquina que por su precio alrededor de 130.000 ptas. se sitúa en la cumbre de la gama y en el que Philips apuesta muy fuerte. Dicho ordenador va a causar auténticos estragos entre el público, que necesita un ordenador para trabajar con algo más de potencia, sin tener que recurrir a ordenadores de otra categoría (obviamente



más caros y de dudosas prestaciones).

La verdad es que, por este precio, tenemos la ventaja de poseer una máquina de 64K *ROM*, 256K *RAM* con función de disco *RAM*,

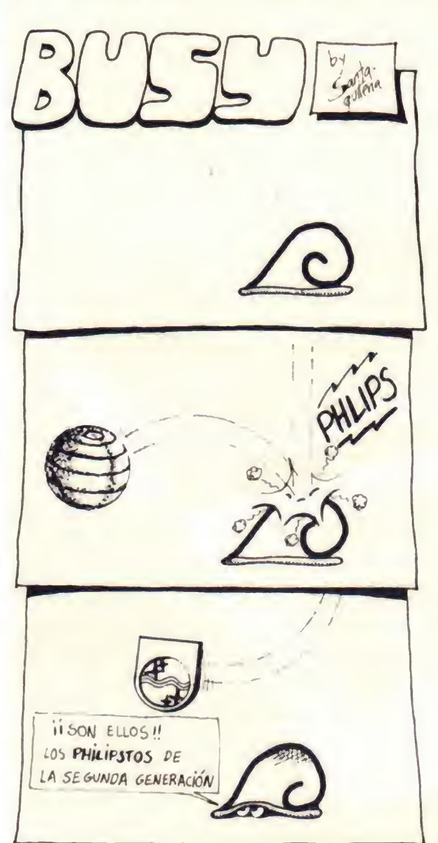
La 2.ª Generación se presenta como un ordenador totalmente semiprofesional.

unidad de disco incorporada de 3.5 pulgadas, 80 columnas, 256 colores, teclado ajustable y tres buses de expansión (dos para cartuchos *ROM* y un tercero para otra unidad de disco). Esto es sólo parte de los que viene con el ordenador. Cabe destacar que, el profano se verá respaldado por diversos manuales, muy completos y simples, y que el experto agradecerá el *software* que tiene para empezar a programar. Dicho *software* consta de tres programas de utilidades, todos en disco, a saber; el sistema operativo MSX-DOS con su manual correspondiente, un paquete de aplicaciones para el hogar *MSX HOME OFFICE* + *MSX DESIGNER* + *DEMONSTRATION SET* y un paquete gráfico con el que podremos comprobar las increibles posibilidades gráficas de este nueva versión, ambos con su manual. Para este último paquete es imprescindible una televisión o un monitor en color, de lo contrario no se podrá apreciar las cualidades gráficas de la máquina.

Sinceramente, creemos que el precio está muy bien ajustado para la diversidad de elementos que el fabricante ofrece con el equipo y dudamos que, hoy por hoy, exista una opción mejor.

Características técnicas

En principio destacaremos que se mantiene la compatibilidad del sistema MSX. Por lo que se refiere a *software* y periféricos, el *VG-8235* y otros ordenadores de la 2.ª Generación gozan de una compatibilidad ascendente con los modelos MSX-I ya existentes. Esto significa que, en principio, todo el *software*



y periféricos de los MSX-I funcionarán con el MSX-II, lo contrario no es necesariamente cierto, ya que determinado *software* MSX-II utiliza algunas características del *hardware* que no existen en las versiones anteriores (como por ejemplo la disco *RAM*).

Esta compatibilidad, en principio se debe a que mantiene el microprocesador de las otras versiones, es decir, el *Z-80A*. Con ello, no se gana en rapidez, pero se mantiene la característica más importante, la compatibilidad.

La capacidad de memoria aumenta considerablemente, pasando la memoria *RAM* a poseer 256K (128K de vídeo y 128K de usuario). Esto es posible gracias al nuevo chip de vídeo *V9938*, un circuito integrado *VLSI* con 64 *pines*, compatible totalmente con el *TMS9918A* y bastante más potente.

El cambio de chip ha traído consigo nuevos aires a la representación en pantalla, con más posibilidades gráficas, más colores, más *sprites*, etc., que ya comentaremos más adelante.

La *ROM* de 64K, está dividida entre los 48K para el MSX-II y los 16K para control de las unidades de disco, con más instrucciones, unas ampliadas y otras nuevas, como por ejemplo, *WIDTH* la cual permite la modalidad de 80 columnas sin necesidad de poseer algún cartucho de expansión. Estas son las instrucciones que se han visto ampliadas:

SCREEN, SET PAGE, WIDTH, LOCATE, COLOR, COLOR SPRITE, COPY, CIRCLE, LINE, PAINT, PSET, VDP, SET VIDEO, VPEEK, VPOKE, BSAVE, BLOAD, BASE, GET DATE / TIME, SET DATE / TIME, SET ADJUST / BEEP / TITLE / PASSWORD / PROMPT / SCREEN, nombre MEM, CALL MEMINI / MFILES / MKILL / MNAME y PAD.

También se ha cambiado el chip de sonido. Sin embargo, el nuevo chip *S-3527* (totalmente compatible con el *AY-3-8910*) po-

see las mismas características que aquel, 8 octavas y 3 acordes de tonos.

Hasta aquí, podemos comprobar los importantes cambios que se han realizado en el ordenador, pero, sin lugar a dudas el punto más fuerte de esta máquina reside en la capacidad gráfica. Ahora no hay 16 colores, sino un total de 256 o la posibilidad de combinar 16 de una paleta de 512. El número de colores depende del modo de pantalla en que se encuentre el ordenador. Estas se han visto incrementadas a 9 tipos de *SCREEN* (desde el 0 al 8) distintos, que están divididos en 2 modos de texto y 7 modos gráficos. En el primer modo de texto, se podrá utilizar el *software* de la primera generación, puesto que es compatible con el chip *TMS9918A*. Sin embargo, el segundo modo de textos no permite usar el *software* antes mencionado debido a la utilización de 80 columnas, algo que antes se conseguía con el cartucho *ROM* correspondiente.

Es obvio que en este punto no hay competencia de ninguna clase. Pero aquí no queda esto y los *sprites* así como su manejo se han visto alterados. En las versiones anteriores, sólo se podían representar un total de 4 *sprites* en línea, con los consiguientes problemas a la hora de tener que poner más. El nuevo procesador de vídeo, permite representar un total de 8 *sprites*, pero se pueden definir hasta 256, esto facilita la creación de innumerables aplicaciones, desde juegos a programas de gestión.



Desde el punto de vista de la programación, las instrucciones se han ampliado (como hemos visto anteriormente), a la vez que se han creado algunas nuevas. Estas son *SET VIDEO*, para activar la superposición en el modelo previsto para ello, o *CALL MEMINI/MFILES/MKILL/MNAME*, para utilizar el disco *RAM*. En este apartado, existen nuevas instrucciones, tales como *HIGH SPEED MOVE* (avance rápido) o *LOGICAL MOVE* (avance lógico) entre otras. La instrucción *MOVE* puede ejecutarse entre la *RAM* de vídeo y la *RAM* de vídeo, entre un registro y la *RAM* de vídeo y entre la unidad central y la *RAM* de vídeo, y permite realizar operaciones por *bytes* o por puntos.

Todas estas instrucciones tienen su explicación en los manuales que posee el ordenador, con lo que huelga profundizar en el tema, sólo resaltar que todas las instrucciones de gráficos han sido ampliadas y mejoradas para poderse acoplar a las nuevas características del ordenador.

El ordenador

Viendo la máquina, tenemos la sensación de estar ante un aparato mejor preparado que su hermano pequeño.

Sus dimensiones, 420 x 283 x 85 mm (ancho x fondo x alto) y su peso, casi 4 kg, le confieren un aspecto sólido poco frecuente en este tipo de ordenadores.

El teclado tiene un ángulo de ajuste entre 5 y 12 grados, lo que permite adoptar la posición idónea para trabajar. Las teclas tienen el mismo tacto que el *VG-8020*, muy suave pero efectivo. Al principio los dedos vuelan sobre las teclas, al no presentar resistencia y alguna que otra vez daremos a más de una tecla. Las teclas de

Disco RAM y 512 colores son algunas de sus cualidades más importantes a tener en cuenta.

NUESTROS GR
PA
MS

Konami®

COMIC BAKERY

BOXEO

TIME PILOT

SKY JAGUAR

FUTBOL

CIRCUS CHARLEY

GOLF

HYPER RALLY

PING PONG

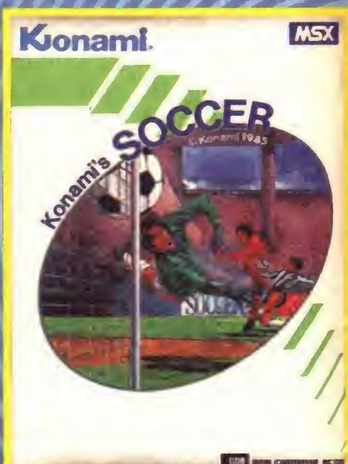


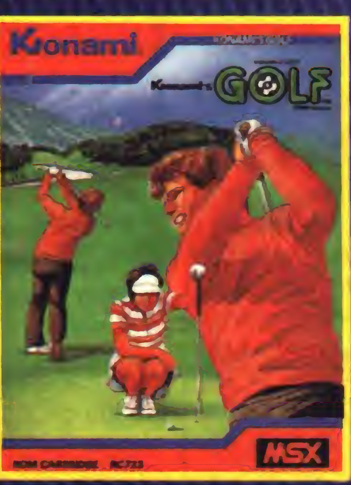


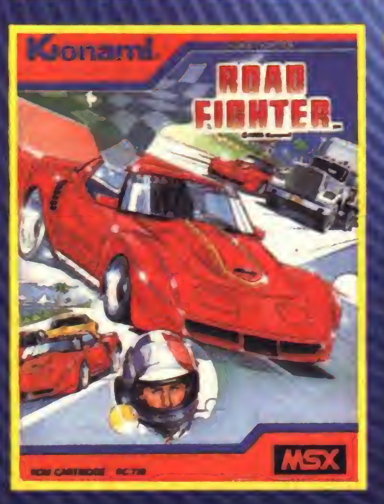














YA A LA VENTA EN TODOS LOS

ANDES EXITOS
RA









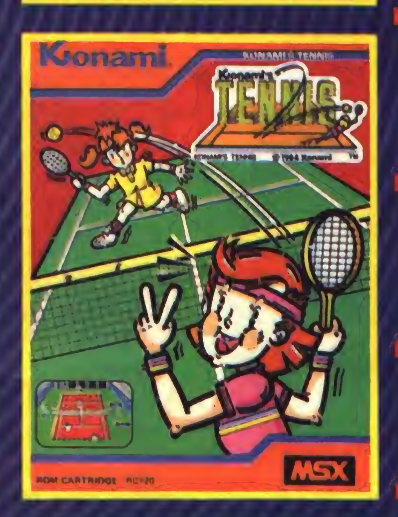










YIE AR KUNG FU

HYPER SPORTS 2

HYPER SPORTS 1

SUPER COBRA

TENNIS

MOPIRANGER

ROAD FIGHTER

YIE AR KUNG FU 2

HYPER SPORTS 3

ESTABLECIMIENTOS DE El Corte Inglés

VG-8235



Foto 1: El VG-8235, presenta un teclado con 73 teclas y un sólido acabado.

función y las de control se encuentran en la parte superior del teclado, son de forma alargada y están dispuestas de manera muy asequible. El *reset* también se encuentra a mano, pero separado del teclado y protegido, para no pulsarlo por error. Al lado, tenemos tres *leds* que indican el estado de la máquina en cada momento, el rojo indica la alimentación, el verde indica cuándo está en modo mayúsculas y el naranja cuándo está funcionando la unidad de discos. Esta se encuentra a la derecha del ordenador, de fácil acceso y muy silenciosa, cualidad que no hemos podido apreciar en otras marcas. Bajo la unidad, encontramos los dos *port* para *joysticks,* lugar más apropiado que en el modelo *VG-8020,* que los tenía en la parte anterior, con lo que no se podía escribir con estos periféricos conectados.

La unidad de discos de 3.5 pulgadas, permite almacenar 360K formateados. Es de simple cara y doble densidad, posee 80 pistas por cara, 9 sectores por pista y 512 *bytes* por sector. La posibilidad de

Un chip de vídeo nuevo permite controlar 128K de VRAM siendo el principal artífice de la increíble resolución que posee.



Foto 2: La unidad de disco de 3.5 pulgadas es la más silenciosa



Foto 3: Vista posterior, en la que podemos destacar un port para cartuchos y un interface para otra unidad de discos.

conectar otra unidad de discos, gracias al interface *FDD* que incorpora para ello, con lo que el sistema se potencia hasta unos límites que rondan la profesionalidad, ya que las dos unidades más el disco *RAM* facilitan enormemente la labor. De cualquier manera habrá que esperar a que lleguen programas específicos para esta generación.

Los interfaces y el resto de los conectores se hallan en la parte posterior del ordenador, donde podremos encontrar dos buses de expansión, uno para cartuchos *ROM* y otro para la mencionada unidad (además de un tercer bus para cartuchos en la parte superior del teclado). También hallaremos el resto de los interfaces de los MSX, Centronics, TV, Monitor, euroconector y la red de alimentación.

La ROM de 64K tiene nuevas funciones entre las que podemos destacar, los comandos que permiten controlar el disco RAM.

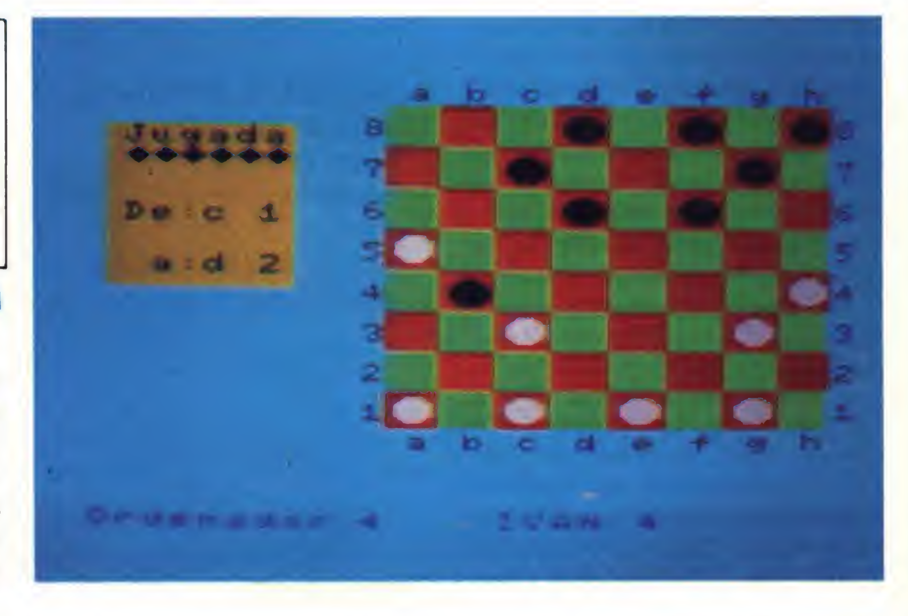
Conclusión

En suma, es un ordenador que viene preparado para afrontar el creciente mercado de los 128K. A pesar de la apretada competencia, estos se las tendrán que ver con uno de los más completos ordenadores personales que hemos visto.

Hoy por hoy, ni el *CPC-6128* de *Amstrad*, ni el *C-128* de *Commodore* y menos el *Spectrum 128*, son enemigos serios a tener en cuenta, aunque sólo destaque uno entre todos, el *Amstrad*. Aún así sólo por la cantidad de *software* que viene con el ordenador y los completos manuales que acompañan a todos estos, el Philips se lleva la palma y a nuestro entender, tiene cualidades que el resto de los ordenadores de la competencia no tiene.

SOFTWARE

Programa: Damas
Tipo: Juego
Distribuidor: Idealogic
Formato: Cassette



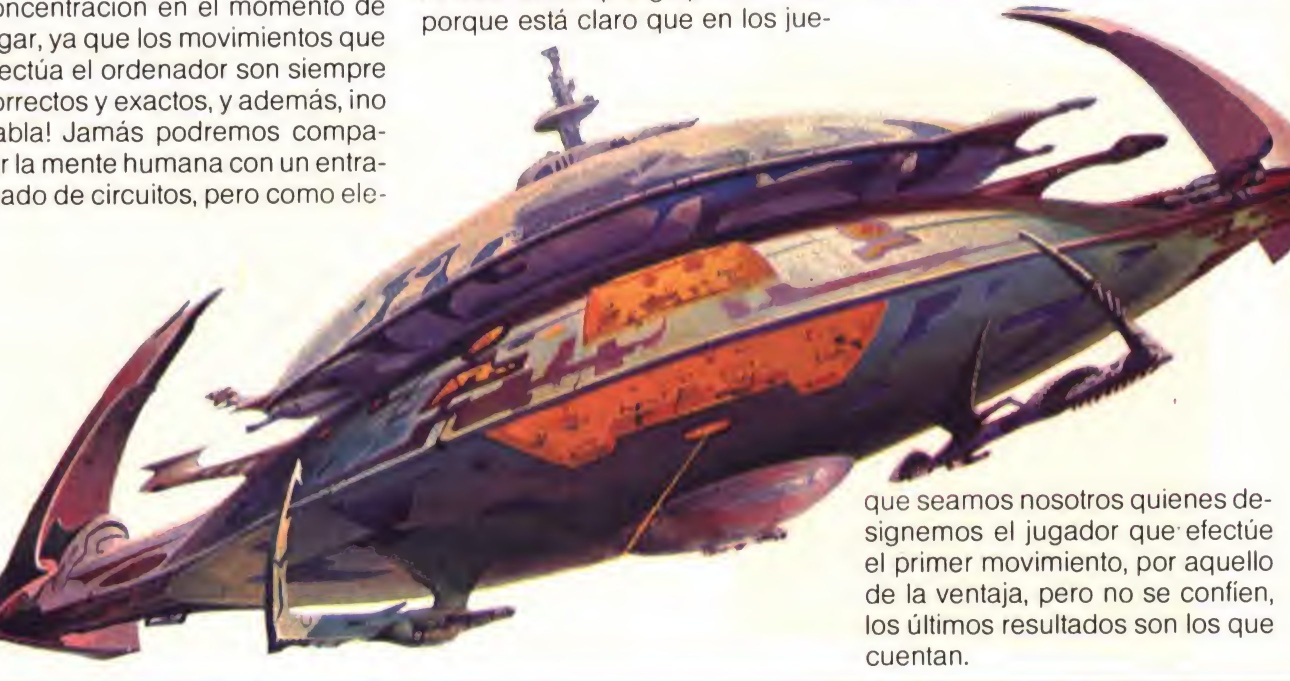
Los clásicos volverán. No es una frase dicha en balde, porque aunque los juegos de todo tipo y color invaden el mercado, siempre nos acordamos de aquellos que se crearon hace tiempo y que aún hoy día perduran, como es el caso de las *Damas*.

Un tablero de damas con sus fichas correspondientes, no puede encontrar mejor sucesor que el monitor de un ordenador, que nos permite contemplar con detallada precisión el color de nuestras fichas y los cuadros por los que podemos movernos sin apenas dificultad.

Damas nos permite una gran concentración en el momento de jugar, ya que los movimientos que efectúa el ordenador son siempre correctos y exactos, y además, ¡no habla! Jamás podremos comparar la mente humana con un entramado de circuitos, pero como elemento auxiliar de compañía es, sin duda, el mejor.

La distribución del juego, aparte de las reglas propias de éste, es sencilla y comprensible para todos, tan sólo es necesario saber jugar un poco a las damas. Los gráficos detallados en su mayor parte nos hacen recorrer la pantalla con un simple golpe de vista, porque está claro que en los juegos de estrategia la anticipación es imprescindible.

En la pantalla podemos encontrar resumidos todos los datos necesarios para poder desarrollar y llevar un control del juego de lo más exacto. Este comienza preguntándonos nuestro nombre, con el que se referirá a lo largo de la partida, después nos permitirá que seamos nosotros quienes designemos el jugador que efectúe el primer movimiento, por aquello de la ventaja, pero no se confíen, los últimos resultados son los que cuentan.



Jugaremos siempre con el color blanco y moveremos nuestras fichas siguiendo las reglas de las damas, ya que si intentamos hacer trampa nos llamará la atención con ¡jugada ilegal!

Una observación a tener en cuenta es cómo debemos darle los datos a la máquina para que efectúe nuestra jugada, el tablero se encuentra numerado en los laterales del 1 al 8 y horizontalmente de la A a la H. Por tanto, si queremos mover una ficha tenemos que dar primero las coordenadas del lugar en que se encuentra y posteriormente el lugar o cuadro en el que queremos colocarla, bien sea para efectuar un mero movimiento o bien para comer una ficha, pero hay que hacer una salvedad en lo que a este apartado se refiere, en el caso de que veamos la posibilidad de comer dos fichas con una nuestra, le indicaremos las coordenadas como si fuéramos a comer tan sólo una y la máquina hará el resto.

Una pequeña advertencia, porque el que avisa no es traidor: se soplan fichas por no comer. No existe espacio limitado de tiempo para realizar una jugada, por lo que aquellos que no sean muy decididos en sus movimientos pueden tener unos momentos de reflexión, además no es bueno precipitarse.

La calidad de jugadas del ordenador alcanza un siete en una escala de diez, por lo que pueden jugar todas las personas principiantes que deseen probar su pericia por primera vez frente al ordenador y también les vendrá bien a los consumados jugadores para que no se olviden de esos pequeños detalles que pueden hacer perder una partida en un momento.

En esos ratos de ocio, ya sabe dónde se encuentra su distracción.

Puntuación:
Presentación: 7
Rapidez: 6
Claridad: 9
Adicción: 8

Programa: Ghostbusters
Tipo: Juego
Distribuidor: Activision
Formato: Cassette

¿Acaso alguien ha visto un fantasma? En realidad sí, gran cantidad de fantasmas se encuentran ahora volando hacia la terrible central de espectros, donde los *Slimers,* nombre con el que se designa a estos fantasmas, se reunen.

En el centro de la ciudad se encuentra la mansión o templo de *Zuul,* señor de los fantasmas, que tiene como propósito destruir la ciudad, y sólo tú puedes prevenir un desastre de proporciones apocalípticas.

Si un día se levanta por la noche y encuentra un fantasma comiéndose su pastel preferido de fresa, ¿qué haría?, llamen a los *Ghostbusters.*

Este programa nos proporciona una aventura sin igual, persiguiendo por los lugares más recónditos estos simpáticos pero peligrosos seres.

¿Qué elementos necesitamos para iniciar nuestro propósito?, por supuesto, no temer a estos pequeños monstruos que se divierten asustando a la gente, pero debemos tener una serie de elementos adecuados para su captura. Un coche potente será nuestro primer objetivo, claro que con nuestros ahorros no podremos aspirar a mucho, por lo que podemos elegir entre cuatro modelos e ir cambiándolos para mejor si nuestro trabajo es eficiente, ya que cada vez que capturemos un fantasma nos recompensará con una suma de dinero. El primer coche es muy



SOFTWARE

pequeño, aunque para empezar nos sirve. El segundo modelo es el que recomendamos, ya que no es muy caro y cabe el equipo necesario. Los otros dos son ya un poco caros para nuestro presupuesto inicial, ya que además debemos comprar otras cosas.

Los elementos de captura son las trampas y la aspiradora de fantasmas (muy útil). Una vez conseguido esto no queda más que salir en busca de ellos.

El juego se divide en tres pantallas, una primera en la que se nos muestra la distribución de la ciudad y donde se encuentra la mansión de *Zuul*, la cual debemos custodiar, para conseguir que los fantasmas no se reunan allí. Utilizar el plano de la ciudad nos vendrá muy bien para poder saber dónde se encuentran los fantasmas y dirigirnos hacia ellos. Una vez que nuestro sensor localice a uno de estos espectros, tendremos que perseguirlos por las calles en las que se encuentran. Este será el momento de probar las características de nuestro automóvil, y una vez que los veamos intentaremos capturarlos con nuestra aspiradora.

Si lo conseguimos pasaremos a otra fase del juego, intentar cazar a un *Slimer* aunque esto resultará mucho más difícil, pues una vez instalada la trampa tendremos que acorralarle con nuestros rayos láser y accionar la trampa.

Cada vez los fantasmas se irán haciendo más rápidos y si logran unirse todos ellos, *Zuul* iniciará su andadura por la ciudad destrozando las casas y todo lo que encuentre a su paso.

Siempre que cazemos a un fantasma conseguiremos dinero que nos permitirá adquirir más elementos de persecución e incluso cambiar nuestro coche por uno mejor, pero también podemos acabar con nuestro dinero y no conseguir nada.

Este juego no es tan sólo una creación de software, sino también una muestra de los avances conseguidos en colores, gráficos y sonido de nuestro ordendor, ya que la melodía con la que se inicia el programa, producida por el generador de sonidos del ordenador, es la réplica exacta de la banda sonora de la película.

Ya sabe, ¿acaso alguien ha visto un fantasma?, *Ghostbusters.*

Puntuación:
Presentación: 8
Rapidez: 8
Claridad: 7
Adicción: 9

Programa: Zakil Wood
Tipo: Juego
Distribuidor: Power Soft
Formato: Cassette

Una historia interminable, una aventura de caballeros, el espíritu de un pueblo, y toda la magia que proporcionan las leyendas, se encuentran reunidas en *Zakiol Wood.*

El maravilloso mundo de la imaginación va asociado, de forma indirecta, con el peligro, la curiosidad y la valentía digna del más astuto guerrero salvador de su pueblo. En el profundo bosque de *Zakil* se encuentra escondido el rubí que ha protegido de catástrofes, tiranías y maldiciones a los habitantes de este pueblo, y nosotros hemos de recuperarlo. Viviremos y descubriremos toda clase de peligros, pero conseguiremos la victoria.

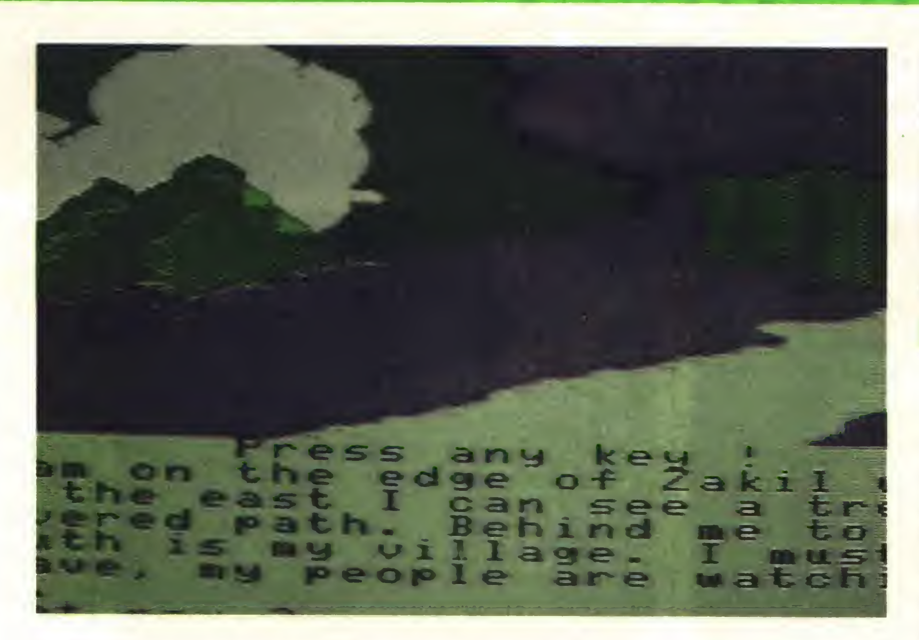
La aventura comienza si conseguimos pasar por el *Pyral* de dos cabezas, templo donde las fuerzas malignas y la magia se reunen, pero antes de llegar aquí, debemos equiparnos con todos los elementos necesarios para enfrentarnos a monstruos y situaciones inverosímiles. Estos objetos los iremos encontrando a lo largo de la aventura, pero debemos escoger bien, ya que sólo podremos transportar un número limitado de ellos.

No siempre la fuerza lleva a la victoria, hay que emplear la inteligencia y, sobre todo, elegir el camino que nos llevará al rubí, pues más de una vez nos encontraremos en encrucijadas de caminos y no sabremos hacia dónde dirigirnos.

Este juego no sirve sólo para divertirnos, ya que las instrucciones están escritas en inglés y los términos que empleemos para desarrollar la aventura, también deben ir en inglés. Esto no es problema,



ya que los términos que se utilizan son muy coloquiales y no es necesario componer frases, sino tan sólo palabras concretas cuyo sig-



nificado indique una acción a realizar, como: "GO NORTH" o "GO SOUTH", lo que no resulta problemático y, además, nos sirve para perfeccionar nuestro inglés de la manera más divertida.

Nos encontramos con paisajes sospechosos, como el bosque de *Zakil,* y otros en los que parece que nada ocurre, pero que son los que mayor enigma encierran.

Nuestra espada de caballero, objeto que debemos transportar a lo largo de toda la aventura, nos será un elemento muy útil, aunque existen otros objetos que podemos recoger o desechar, transportando un número limitado de ellos, ya que si los transportamos todos perderíamos nuestras fuerzas.

Si en algún momento queremos saber lo que transportamos, teclearemos "INV" y nos hará una descripción detallada de todo lo que llevamos en ese momento.

Esperamos que esta aventura te hará disfrutar de buenos momentos, sin preocuparte de los peligros y la tensión por los que pases. El color, dentro de los dieciséis colores que nos proporcionan los ordenadores MSX, ha sido realizdo por la configuración de los elementos y la proyección de éstos.

La presentación del programa y la sencillez de su realización, hace de él un buen juego de software.

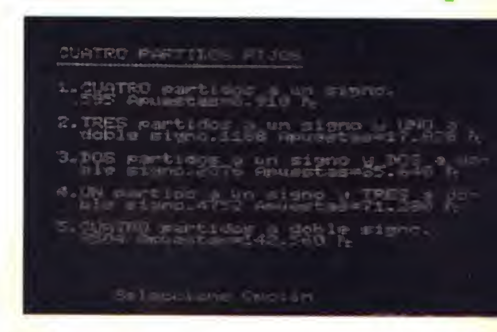
Puntuación:
Presentación: 9
Rapidez: 8
Claridad: 8
Adicción: 8

Programa: 1X2, CON PRONOSTICO
Tipo: Aplicación
Distribuidor: SERMA
Formato: Tarjeta ROM

¿Qué le parece ser millonario?, no pierda esta oportunidad y acierte la quiniela que nunca ha podido realizar, bien por falta de tiempo, o bien porque era muy difícil hacer esas combinaciones sin que se repitiese ninguna apuesta. Ahora con *1X2, CON PRONOSTICO,* encontrará una ayuda para hacer sus sueños realidad.

Es un programa creado con un desarrollo de la quiniela de 4 resultados fijos y 10 a triples reduciéndose en sesenta y dos boletos múltiples, los cuales son el desarrollo de combinar las 10 apuestas triples entre sí.

El proceso de elaboración de la quiniela es muy simple, no es necesario complicarse la vida en saber qué número de boletos ocuparán las apuestas realizadas, o el precio de estos boletos. Partiendo de una quiniela base o «Quiniela de partida», repartiremos un grupo que serán las apuestas múltiples. Aquellos que elijamos po-



PROGRAMAS SONY MSX

Educativos

- Monkey Academy
- Alfamat
- Viaje Espacial
- Multipuzzle
- Noria de Números
- Corro de Formas
- Coconuts
- Yo Calculo
- Selva de Letras
- El Cubo Informático
- Electro-graf
- El Rancho
- Teclas Divertidas
- Boing Boning
- Compulandia
- Mil Caras
- Logo
- Países Mundo-1
- Países Mundo-2
- Tutor
- Computador Adivino
- Aprend. Inglés-1
- Aprend. Inglés-2
- Cosmos
- Curso de Básic
- Juego de Números

Juegos

- Antártic Adventure
- Athletic Land
- Sparkie
- Juno First
- Car Jamboree
- Battle Cross
- Crazy Train
- Mouser
- Computer Billiards
- Alí Babá
- Track & Field-I
- Track & Field-II
- Dorodon
- Chess (Ajedrez)
- Senjo
- E.I.
- Lode Runner
- Super Tennis
- Backgammon
- Super Golf
- Hustler
- Binary Land
- Driller Tanks
- Stop the Express
- Ninja
- Les Flics
- La Pulga
- The Snowman
- Cubit
- Pack 16K
- Fútbol
- Kung Fu
- Batalla Tanques
- Mr. Wong
- Xixolog
- Buggy
- Sweet Acorn
- Peetan
- Jump Coaster
- Buggy 84
- 3D Water Driver
- Pinky Chase
- Wedding Bells
- Fightting Rider

Aplicación

- Memoria Ram 4 K
- Creative Greetings
- Character Collect
- Quinielas y Reducciones
- Pascal
- Ensamblador
- Generador Juegos

Gestión

- Hoja de Cálculo
- Homewriter
- Control Stocks
- Contabilidad Personal
- Ficheros
- Procesador de Textos
- Control Stocks
- Vencimientos
- Contabilidad 1.500

drán variar su resultado bien en 1X, 2X ó 12, que serán propiamente los que varíen, mientras que los demás permanecerán constantes. Los resultados constantes se dividen a su vez en tres grupos, los que pensamos pueden ser partidos difíciles y dentro de una pequeña seguridad, en el último momento pueden variar, un segundo grupo en que la posibilidad de acertar es casi del 90%, los conocemos como los 1 de las quinielas normales. Y el tercer grupo son los fijos, o invariables, aunque en el mundo de los juegos de azar nunca se sabe lo que puede suceder, porque si no a estas alturas todos seríamos millonarios.

A diferencia de otros programas que se eternizaban para hacer esas variaciones, *1X2, CON PRONOSTICO*, nos resuelve el problema tan sólo en unos instantes y no es en absoluto complicado ver las combinaciones, pues aparecen en el formato de la quiniela ocho columnas, estando muy definidos los carácteres.

Haremos una breve indicación para aquellos que jueguen a cuatro partidos fijos a signo o a doble signo, con cinco opciones variando el número de apuestas y el coste de la quiniela:
595 apuestas = 8.925 ptas., etc.

Con este programa podrá invertir su dinero con un margen de probabilidad superior al que tenía anteriormente, un 10% para el 14, el 42,5% para el 13 y un 67,5% para el doce.



El funcionamiento del juego comienza con la elección de los múltiples que queremos jugar:

— Un doble y tres variables.

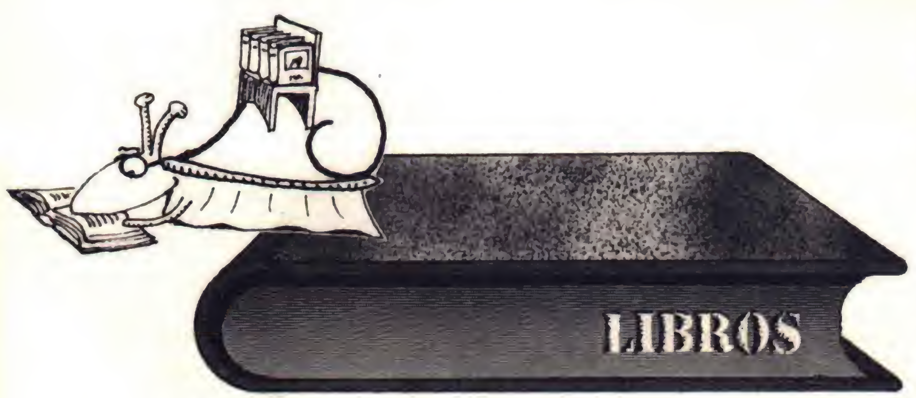
— Dos dobles y dos variables, etc, hasta cuatro.

Una vez realizada la elección pasaremos a completar la quiniela de partida. Nos aparecerán los 14 números en la parte izquierda del monitor y a la derecha los números múltiples que vamos a emplear.

Si se diese el caso que fuesen cuatro los múltiples, tendríamos que ir uno por uno seleccionando las variables que vamos a utilizar.

Acabamos de comentarle las posibilidades de este juego que sin duda le ayudará a resolver sus problemas quinielísticos, pero la última palabra la tiene la suerte.

Puntuación:
Presentación: 8
Claridad: 9
Rapidez: 9
Adicción: —



Título: Lenguaje máquina para MSX, introducción y conceptos avanzados
Autor: Joe Pritchard
Editorial: Anaya Multimedia
Páginas: 240

Programar en Código Máquina supone un aprovechamiento total de las posibilidades de cualquier ordenador y del microprocesador *Z-80*, ya que este es el corazón de todos los MSX. Un programa escrito en Código Máquina puede ejecutarse hasta 20 veces más rápido que su equivalente en *BASIC*, lo cual demuestra la utilidad y lo importante que resulta dominar la programación en lenguaje ensamblador.

Los libros de programación en este lenguaje se están multiplicando por momentos, pero hasta la fecha no habíamos tropezado con uno tan completo como el que nos ocupa.

Este libro, es un curso de programación en lenguaje máquina (ensamblador), orientado tanto a usuarios que no hayan oído hablar nunca de éste, como a los que conociendo la programación del *Z-80* deseen ampliar sus conocimientos sobre el sistema MSX.

Comienza haciendo una descripción de la *CPU (Z-80A)*, así como del resto de los circuitos importantes del MSX: *VDP* (procesador de vídeo), *PSG* (procesador de sonido) y *PPI* (interface periférico programable). A continuación encontramos un capítulo dedicado a los sistemas de numeración y a la utilización de programas en Código Máquina desde el *BASIC*, algo que se debería utilizar con más frecuencia de la usual, principalmente debido a que se mejoran las prestaciones de cualquier tipo de programas.



El resto está dedicado al estudio detallado de cada una de las instrucciones del ensamblador, dividiéndolas en grupos según su función. En esta parte, también se estudia la forma de acceso a los diversos circuitos auxiliares. Todos los capítulos están ilustrados con ejemplos prácticos que posteriormente se podrán incluir como subrutinas en otros programas.

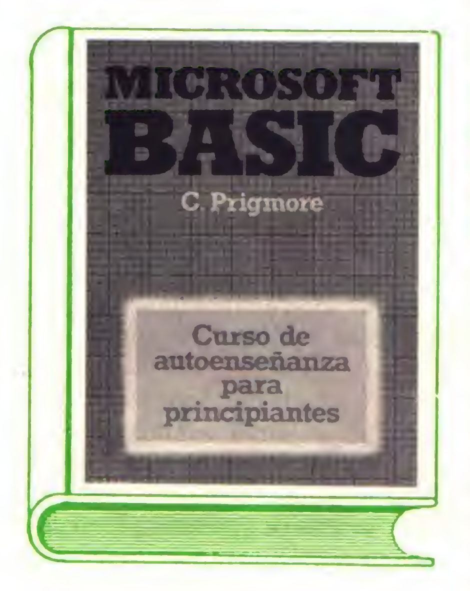
Concluye el libro con una serie de apéndices muy útiles entre los que se incluyen las tablas de conversión decimal-hexadecimal, listado de las instrucciones del *Z-80* y un programa para introducir Código Máquina desde *BASIC*. Este programa será de gran utilidad para todos aquéllos que no dispongan de un ensamblador para introducir los códigos de las instrucciones en la memoria del ordenador. En definitiva, se trata de un libro muy útil, tanto para los que desean aprender como para aquellos que quieran conocer a fondo las características del aparato. La presentación y traducción es buena como viene siendo habitual en las publicaciones de esta editorial.

Título: Microsoft BASIC
Autor: C. Prigmore
Editorial: Gustavo Gili
Páginas: 212

El *BASIC* es un lenguaje de programación con diversos dialectos. Cada ordenador tiene su versión particular, realizada por su fabricante para un modelo determinado. Pero sin lugar a dudas, el más utilizado dentro de su clase es el de *Microsoft*. Grandes empresas, como IBM, utilizan este *BASIC* para sus PCs y MSX lo adoptó para sus ordenadores personales. De hecho, las siglas del estándar significan *Microsoft Superextended BASIC*. Hay literatura sobre el tema, pero ninguna escrita a un nivel tan sencillo y agradable como éste.

Se trata de un libro pensado pa-

ra personas que utilizan por primera vez un ordenador. Por este motivo, el libro comienza explicando los principios básicos de una computadora, así como sus elementos principales. Después de esta breve introducción, continúa con las explicaciones básicas del *BASIC* y los fundamentos de la programación.



El resto del libro está dedicado a examinar cada uno de los comandos y funciones del lenguaje. Todos los capítulos tienen una estructura similar. Comienzan explicando los diversos temas acompañados de numerosos ejemplos prácticos y finalizan con una serie de ejercicios para que el lector aprenda, de una forma amena y práctica, todos los temas tratados.

Merece la pena destacar que solamente se limita a explicar las instrucciones y funciones del *BASIC* estándar, sin complicarse con instrucciones particulares de un ordenador concreto, por lo que resulta válido para los usuarios de todo tipo de ordenadores personales que utilicen este tipo de lenguaje.

En resumen, se trata de un libro ideal para todas aquellas personas que desean introducirse en el mundo de la informática y concretamente en la programación, sin complicarse demasiado.

Una característica muy atractiva que posee, es que todos los programas incluidos en él se presentan resueltos, por lo que no es imprescindible disponer de un ordenador para poderlo utilizar.

Título: El Gran Libro de los programas en BASIC
Editorial: Ediciones Generales Anaya
Páginas: 116

Es indudable que la mejor forma de aprender el difícil arte de la programación es practicando y realizando uno mismo sus propios programas. Pero para ello ha tenido que existir una etapa previa de aprendizaje en la que se habrán estudiado problemas ya resueltos para, en un momento determinado, aplicar el mismo método de resolución a un problema similar. Para ello contamos con diversas publicaciones y libros dedicados a exponer problemas y a resolverlos. ¿Quién no habrá oído o incluso utilizado la colección de *Schawn* de problemas matemáticos, físicos, etc., para resolver sus dudas?

El *Gran Libro de los programas en BASIC,* pertenece a esa línea de publicaciones dedicadas a la complicada tarea de enseñar. Contiene numerosos programas analizados línea a línea, para que cada uno pûeda realizar las transformaciones necesarias o las que a cada cual se le ocurran. El libro está preparado para ser utilizado por los usuarios de cinco ordenadores diferentes: MSX, Vic 20, Spectrum, BBC y Dragón. Los ocho capítulos en que se divide, se pueden agrupar en varias secciones. Una primera, de tres capítulos, que está dedicada a la descripción de las rutinas más importantes de cada ordenador y luego una sección por capítulo, ya que cada uno de ellos trata un tipo de programa particular. En estos capítulos hay un total de 48 pro-



gramas comentados a fondo, con sus características más interesantes, lo que permite realizar los cambios oportunos. En estos capítulos, el lector encontrará programas de diversión, de pensar, rompecabezas de números, problemas científicos y problemas de palabras.

El libro es ideal para los que inician su andadura informática, y para los pequeños de la casa, a los que atraerán los juegos más que otra cosa. Además, las numerosas ilustraciones a todo color que acompañan al texto, hacen ameno y comprensible el tema que está tratando. Por otra parte, los listados de los programas no son excesivamente largos, con lo que se evitan muchos errores y se anima al lector a que los teclee cuidadosamente.

Los MSX de Ya[...]

CX-[...]



Foto 1: Los Yamaha son ordenadores con unas prestaciones fuera de toda duda.

Cada fabricante de MSX nos tiene acostumbrados a un detalle que hace de su ordenador un equipo especial. Es el caso del PIONEER que puede trabajar con sistemas de vídeo, del SONY que incluye una agenda en ROM, etc. En el caso del YAMAHA nos encontramos con un potente sintetizador de FM. No podía ser otra que una prestigiosa casa de instrumentos musicales la que pusiera el estándard al servicio de la música.

Dos son los ordenadores con los que cuenta: El CX-5M y el CX-5M II. La única diferencia que separa a uno de otro, es que el segundo posee 64K RAM, frente a los 32K del primero, y dos conexiones para cartucho, mientras que el primero sólo posee una y un slot de expansión, que puede ser adaptado para aceptar cartuchos mediante un simple accesorio. El CX-5M II incorpora un pulsador de RESET y una salida para monitor RGB. El teclado ha sido ligeramente reformado, siendo las teclas de función más anchas. Como MSX son prácticamente iguales que el resto, pero lo que les hace realmente distintos es el sintetizador y el interface MIDI que lo acompaña.

El sintetizador

Se trata de un módulo que el ordenador incorpora a un lado, y que se separa simplemente por un tornillo, si bien su ensamblaje es robusto y fiable. Basta con encender el ordenador y llamar a un programa incorporado en ROM (call music) para tener en nuestras manos el control del sintetizador. Podemos dividir el teclado musical en dos partes, y asignar a cada una un sonido diferente entre 46. Una de las partes será polifónica y la otra monofónica, siendo posible su inversión en todo momento. En el programa music del CX-5M II el teclado se puede dividir esta vez en dos partes polifónicas (de 4 voces cada una). El nuevo programa posibilita además la mezcla de dos sonidos en uno (también polifónico a 4 voces); en esta opción son controlables los volúmenes de mezcla de cada sonido y la desafinación entre ambos, con lo que se pueden enriquecer mucho los diferentes timbres. Ha sido perfeccionado respecto del control sobre el MIDI, lo que en la primera versión no era posible. Finalmente, se ha incluido la mejora de trabajar con disco (aparte del cassete o del cartucho RAM) que en principio no existía.

Tiene un secuenciador en tiempo real de 1800 notas. Es decir, es suficiente con pulsar la tecla de grabación y todo lo que toquemos será grabado en la memoria, hasta que pulsemos STOP (fin de grabación). Puede crear acordes y bajos automáticos y tenemos la opción de escoger entre seis ritmos, para que si queremos una batería electrónica (de no mucha calidad) nos acompañe.

Aparte de estas posibilidades que el CX-5M tiene integradas en sí mismo, podemos añadir otras muchas con los cartuchos creados específicamente para este ordenador. Básicamente son:

Voicing-Program: Es un progra-

maha

-5M y CX-5M II

volventes que nos permite controlar la evolución en amplitud de la onda. Posee un LFO (Low Frecuency Oscillator/Oscilador de baja frecuencia) que sirve para modular en frecuencia a los operadores, y producir, entre otros, efectos de vibrato, o modular en amplitud para dar lugar a trémolos, etc.

Los nuevos timbres que creemos podrán ser almacenados en cinta o disco (en la nueva versión) para su uso posterior, bien en el modo sintetizador o con otros programas.

Se pueden sacar copias de pantalla por impresora (bien MSX o formato Epson) para comparar o estudiar los parámetros de diversos sonidos. Este programa va incorporado en un pequeñísimo cartucho en el CX-5M II (Voicing Program II). En el manual de este cartucho se incluye un programa en BASIC que nos mostrará cual es el resultado de modular en frecuencia a un operador, con otro de frecuencia y amplitud a elegir, o bien la composición aditiva de ambos operadores.

DX 7 Editor: Es un cartucho creado para visualizar los parámetros del sintetizador DX 7 de Yamaha haciendo más cómoda su modificación, creación de soni-

ma que nos da acceso al manejo de los parámetros del sintetizador (más de 70). Este es del tipo FM (Modulación en Frecuencia). Está compuesto por cuatro osciladores u operadores —generadores de onda sinusoidal—' que se pueden combinar de ocho formas diferentes (algoritmos) creando por tanto un gran número de timbres. Un operador puede funcionar como modulador, es decir, modular en frecuencia a otro operador, o como portador, del cual se toma la señal de salida. Sólo un operador se puede modular a sí mismo mediante la realimentación o Feedback (operador número uno). Es posible crear frecuencias que pierdan la relación con los otros operadores, para producir disonancias, sonidos de campana, etc. Cada uno de los operadores va asociado a un generador de en-



Foto 2: El CX5M II, versión mejorada de su hermano pequeño

dos y observación de los gráficos de las formas de las envolventes, al igual que el anterior son posibles copias por impresora.

Music Composer: Este programa es un secuenciador paso a paso de ocho pistas, en las que podremos grabar diferentes partes de un tema con diversos instrumentos y matices de interpretación, distribución de los sonidos en el canal deseado (izquierdo-derecho), etc. Convertirá al CX 5M en una eficaz herramienta para ser usada por músicos, compositores y arreglistas. Pueden sincronizarse o ser controlados otros instrumentos via MIDI.

Este cartucho también tiene su sucesor, el Music Composer II, que permite almacenar los datos en disco, entre otras mejoras.

MIDI Recorder: El MIDI (Musical Instrument Digital Interface) no es otra cosa que un interface que hace que los diferentes instrumentos que lo tengan intercambien información musical, interna, etc. En cuanto a hardware es un transmisor receptor universal asíncrono (UART) que trabaja a 31.25 Kbaudios. Utiliza diez bits en la transmisión en serie de datos: el primero de comienzo, ocho de datos, y uno de parada. Una serie de mensajes son reconocidos por cualquier sistema que incorpore este interface; otros, sin embargo, son transmitidos entre aparatos del mismo tipo, o entre ordendores (con el software apropiado) y estos instrumentos. Son los que conforman el sistema exclusivo de cada fabricante y producto (que va acompañado por un código en hexadecimal que lo identifica). Pues bien, el MIDI Recorder es un secuenciador en tiempo real que reconoce esta información y la graba en memoria. Si utilizamos un teclado externo con MIDI y lo conectamos al CX-5M, grabará la interpretación que efectuemos, incluyendo variaciones en la afinación, vibrato, dinámica (si el instrumento envía este código), etc.

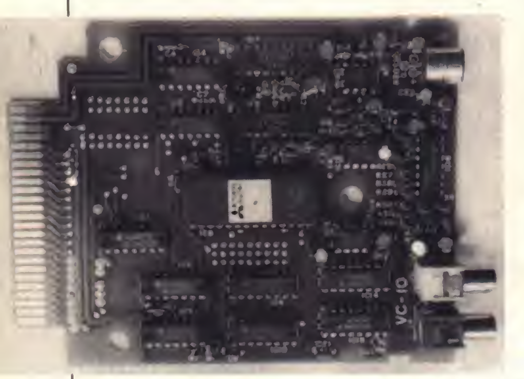


Foto 3: Impresionante perspectiva del sintetizador que

Luego, como si de manejar un magnetofón se tratara, podremos sumar pistas, corregir partes, grabar encima, etc. La capacidad de este secuenciador es de más de 50.000 notas/datos, con lo que se convierte en uno de los más potentes del mercado. Esta capacidad puede ser aumentada si se incrementa la memoria del ordenador.

Ficha Técnica: Sintetizador.

Generador de sonidos: FM 4 operadores. 8 algoritmos.
Polifonía: 8 voces.
Voces internar: 48 (46 pregrabadas).
Conexiones: Salida audio (L/R). MIDI (IN/OUT). Teclado musical.
Salida audio: –9 dBm (440-880 Hz). Sonido FLUTE (flauta). Simultánea 8 voces.
Impedancia: 1.8 KΩ
Margen de temperaturas: 0–35°C.
Humedad: 20–80%
Consumo aproximado: 30 W (máximo).



RX Editor: Como el DX 7 Editor, es un programa muy concreto para trabajar con cajas de ritmo de la serie RX de Yamaha. Mejora las posibilidades de éstas, al darnos la oportunidad de matizar con diez volúmenes distintos en cada instrumento y aumentar la capacidad de su memoria, visualizar los patrones de ritmo creados, etc.



Foto 4: Vista de los

Music Macro y Music Macro II: Son cartuchos para controlar el sintetizador desde el BASIC a través de nuevos comandos, para de esta forma, dar mayor espectacularidad al sonido de nuestros programas. Sustituye de forma incomparable al famoso chip de sonido incluido en los MSX.

Guitar Chord Master y Keyboard Chord Master: Son dos programas destinados a la formación musical. Enseñan, bien sobre un

mástil, bien sobre un teclado, cuáles son las posiciones que se han de adoptar en más de 300 acordes distintos. El segundo incorpora un metrónomo, con posibilidad de acentuar cada vez que pasen el número de partes elegidas.

Graphic Artist: Este programa está destinado a trabajar con un ratón (del que también dispone Yamaha). Convierte al MSX en un gran editor de gráficos, útil para el diseño de pantallas y otras aplicaciones.

En *periféricos,* Yamaha dispone de:

Una *impresora* de matriz para hacer copias de pantalla de muchos de los programas. Acepta papel continuo perforado o cualquier otro tipo de papel, folios, etc. por fricción. Dos son los tipos fundamentales de letra: Pica, 80 carácteres por línea, y élite, 96 CPL. Es bidireccional, con una velocidad media de 40 carácteres por segundo.

Unidad de disco de 720 Kbytes (tras el formateado) con una velocidad de transferencia de 250 Kbits/s. Trabaja con discos de doble cara de 3.5 pulgadas (si bien acepta los de simple cara dividiendo por dos la capacidad.

Un *ratón* que puede funcionar bien como cursor (8 direcciones) o como contador. Para ambos usos se incorporan dos pequeños programas en el manual que nos muestran sus posibilidades.

El ordenador viene acompañado de un completo manual de BASIC MSX, en el que se incluye una introducción al manejo del MSX-DOS, y MSX Disk BASIC. Otros capítulos se dedican al editor (tan poco conocido), al manejo de variables, a las BIOS (Basic Input Output Stament), a la creación adecuada de programas y control de las interrupciones, etc.

Definitivamente Yamaha ha sido la primera en unir en un todo compacto dos estándar mundiales: uno de ordenadores MSX y el otro de instrumentos musicales MIDI. Sólo queda esperar su representante de la segunda generación, que será posiblemente el CX-7.

José Luis Crespo Dueñas

NOTA: El distribuidor de Yamaha en España es HAZEN, Ctra, de la Coruña, km. 17,200. Las Rozas (Madrid).



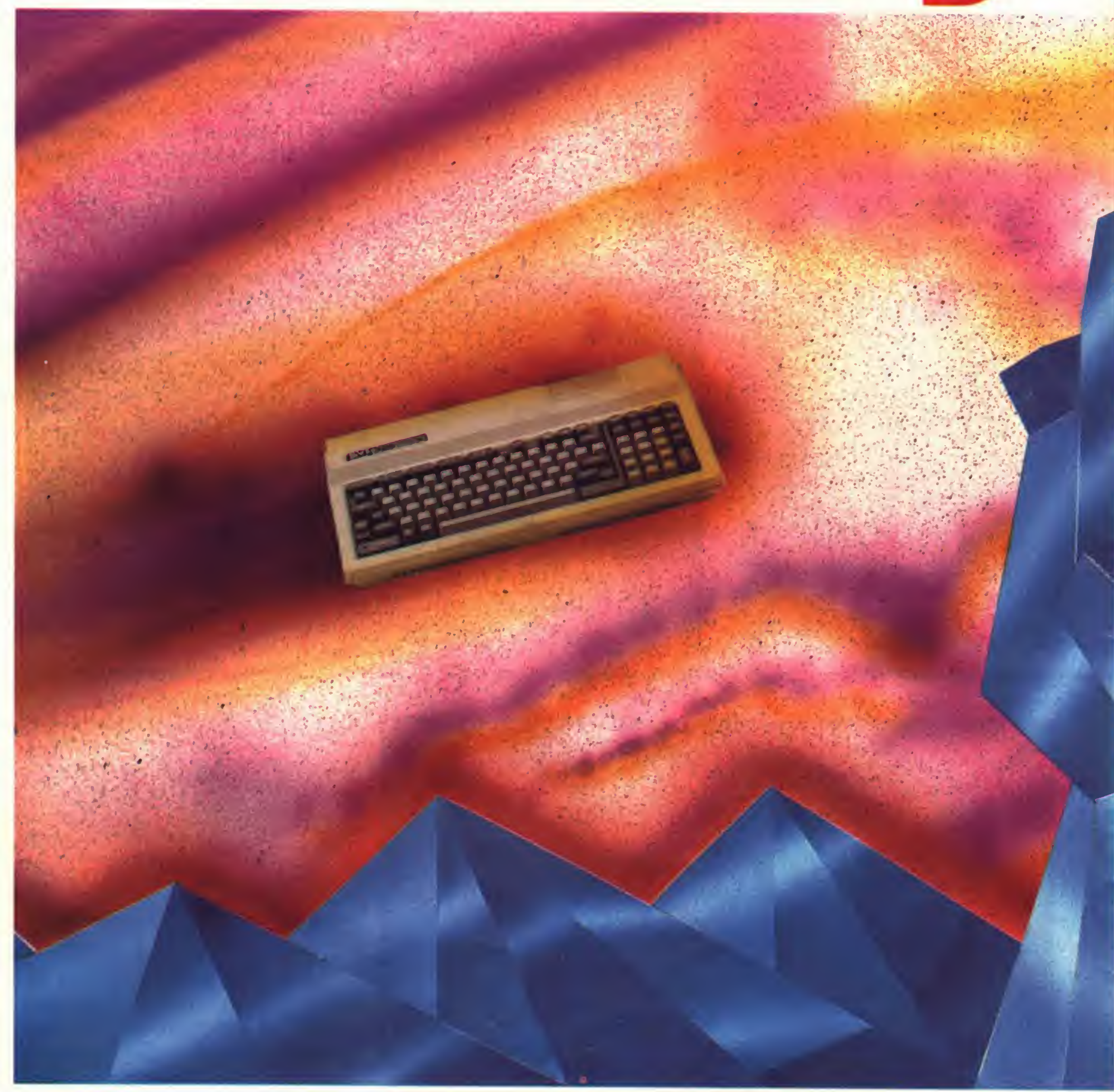
conectores estándar del CX5M.



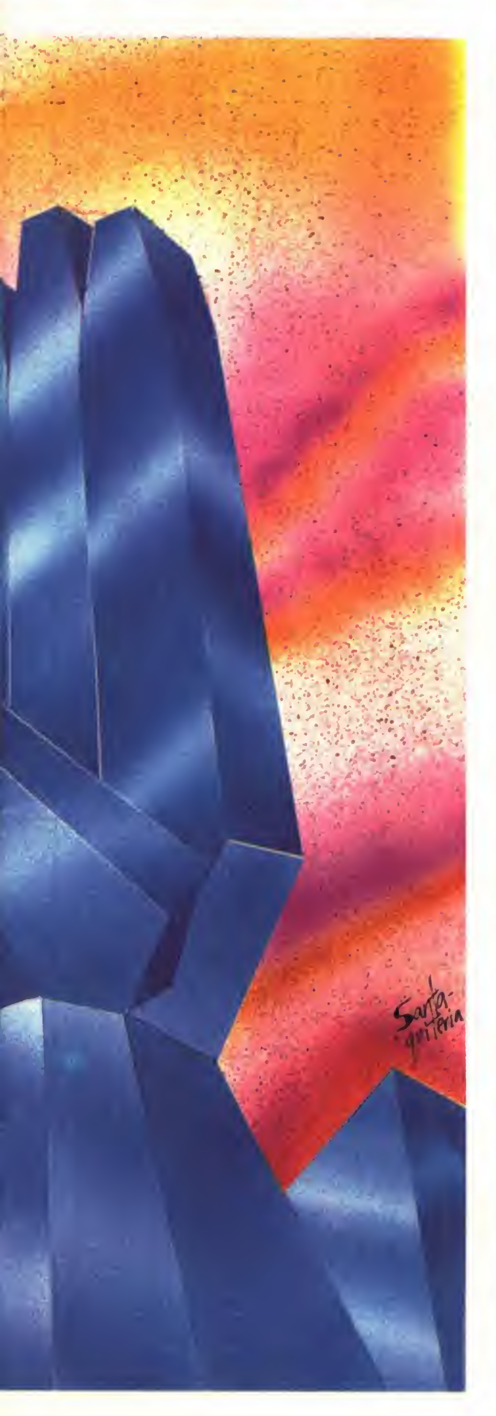
Foto 5: Vista posterior del CX5M II, donde podemos destacar el juego completo de conectores e interfaces.



SVI-318 y 3



328



Continuando con el tema iniciado el mes anterior, vamos a introducirnos en las características internas de estos ordenadores, analizando cómo se distribuye la memoria, así como la función de determinadas zonas.

Area del programa BASIC

Esta zona contiene el listado del programa *BASIC* codificado. La codificación empleada es diferente según que la información a codificar sea numérica, alfanumérica, o un comando o función.

El formato general de cada línea del programa se muestra en la figura 4.

Los dos primeros números de la línea *BASIC*, indican la dirección en el área del *BASIC* en la que comenzará la codificación de la siguiente linea. Estos dos primeros *bytes* no se muestran en la pantalla al hacer el listado de un programa *BASIC*.

Los bytes 3 y 4 contienen el número de línea, esto quiere decir que el número de línea siempre ocupará 2 *bytes*, tanto si el número de línea es un número elevado, como si es un número bajo.

El *byte* final de cada línea es un cero.

Entre el número de línea y el cero final se distribuyen los diversos comandos y operandos que componen una línea.

Los comandos y funciones se representan por un código de uno o dos *bytes* de acuerdo con las tablas de las figuras 5 y 6.

Los números se guardan en forma binaria y el texto en código *ASCII*. En el apéndice A del manual hay una tabla con este código.

Comparando dicha tabla con la de la figura 5 podéis comprobar que los huecos de la tabla *ASCII* están ocupados por la tabla de comandos y funciones.

Si no encontráis la tabla *ASCII* por cualquier circunstancia, podéis sacarla directamente del ordenador introduciendo el siguiente miniprograma:

```
10 FORA=0TO255:PRINTA;
20 IF(A>32ANDA<127)OR(A>159ANDA<212) THENPRINTCHR$(A)ELSEPRINT
30 NEXTA
```

Area de la tabla de variables

Esta tabla guarda información sobre las variables usadas en el programa *BASIC*.

La forma en que se guarda la información difiere si se trata de variables numéricas o de variables alfanuméricas o de cadena sola-

Formato de las líneas BASIC en memoria

NUMERO DE BYTE	CONTENIDO
1 - 2	Dirección de inicio de la siguiente línea
3 - 4	Número de línea
	Comandos y operandos
Ultimo	Cero

Figura 1

mente se guarda la descripción de dicha cadena.

Pero veámoslo más detenidamente:

— Variables numéricas:

El espacio de memoria ocupado por este tipo de variables depende de la precisión con que han sido definidas.

A=, cuya característica es ir seguida por el símbolo «=», es una variable del tipo llamado de «doble precisión». También puede quedar definida permanentemente por *DEFDBLA* con lo que no hace falta que vaya seguida por el símbolo «=» cada vez que se use.

Este tipo de variables puede guardar un número de hasta 14 dígitos. Si se intenta guardar un número con más dígitos, automáticamente el ordenador lo redondea a la cantidad más próxima de 14 dígitos y lo guarda en forma exponencial.

Ej.:

A= = 123456789012345

queda guardado como

A= = 1.2345678901235E+14

que significa que

A= = 1.2345678901235 * 10 ^14

En la tabla de variables estos números de doble precisión ocupan 11 *bytes*.

Tabla de comandos y sus códigos

AND	248	ATTR$	233	AUTO	169
BEEP	192	BLOAD	205	BSAVE	206
CIRCLE	188	CLEAR	146	CLICK	200
CLOAD	155	CLOSE	180	CLS	159
CMD	215	COLOR	189	CONT	153
COPY	214	CRSLIN	232	CSAVE	154
DATA	132	DEF	151	DEFDBL	174
DEFINT	172	DEFSNG	173	DEFSTR	171
DELETE	168	DIAL	208	DIM	134
DRAW	190	DSKI$	234	DSKO$	209
ELSE	161	END	129	EQV	251
ERASE	165	ERL	225	ERR	226
ERROR	166	FIELD	177	FILES	183
FN	222	FOR	130	GET	178
GOSUB	141	GOTO	137	GO TO	137
IF	139	IMP	252	INKEY$	236
INPUT	133	INSTR	229	IPL	213
KEY	199	KILL	212	LET	136
LFILES	187	LINE	175	LIST	147
LLIST	158	LOAD	181	LOCATE	216
LPRINT	157	LSET	184	MAX	202
MDM	207	MERGE	182	MOD	253
MON	203	MOTOR	204	NAME	211
NEW	148	NEXT	131	NOT	224
OFF	235	ON	149	OPEN	176
OR	249	OUT	156	PAIN	191
PLAY	193	POINT	237	POKE	152
PRESENT	195	PRINT	145	PSET	194
PUT	179	READ	135	REM	143
RENUM	170	RESTORE	140	RESUME	167
RETURN	142	RSET	185	RUN	138
SAVE	186	SCREEN	197	SET	210
SOUND	196	SPC	223	SPRITE	238
STEP	220	STOP	144	STRING$	227
SWAP	164	SWITCH	201	TAB	219
THEN	218	TIME	239	TO	217
TROFF	163	TRON	162	USING	228
USR	221	VARPTR	231	VPOKE	198
WAIT	150	WIDTH	160	XOR	250
>	240	=	241	<	242
+	243	—	244	*	245
/	246	>	247		

Figura 2

El primero de estos *bytes* es un 8 para indicar que el número representado es de doble precisión.

Los dos *bytes* siguientes corresponden al nombre de la variable.

Los 8 *bytes* finales guardan el valor.

A!, o la definición previa *DEFSNGA* define a A como una variable de simple precisión.

Puede contener un número de hasta 6 dígitos, redondeando los números que superen esta cantidad, aunque no se presentan en forma exponencial hasta que no sobrepasan los 14 dígitos.

Ocupan 7 *bytes* en la tabla de variables.

El primer *byte* es un 4 para indicar simple precisión.

Los dos *bytes* siguientes codifican el nombre de la variable.

Los cuatro *bytes* finales indican el valor.

A%, o la definición previa *DEFINTA* define a A como variable entera y puede contener números entre –32768 y 32767.

Se representa en la tabla de variables mediante 5 *bytes.*

El primero de estos *bytes* es un 2, lo que indica que lo que va a continuación es una variables entera.

Los dos siguientes *bytes* guardan codificado el nombre de la variable.

Los dos *bytes* finales contienen el valor de la variable entera.

– Variables alfanuméricas.

Como ya mencionamos antes, en la tabla de variables no se guarda exactamente el literal o cadena de carácteres que contiene la variable alfanumérica, sino solamente una descripción de la misma.

A$, o la definición previa y permanente *DEFSTRA* define a A como variable alfanumérica o de cadena, y su contenido ha de ir escrito entre comillas.

En la tabla de variables se describe a las variables alfanuméricas siempre mediante 6 *bytes.*

El primero de los seis *bytes* es un 3 para indicar variable de cadena.

Los dos *bytes* siguientes guardan el nombre de la variable.

El siguiente *byte* indica el número de carácteres que forman la cadena.

Los dos *bytes* finales son correlativamente el *byte* bajo y el *byte* alto de la dirección de memoria donde se puede encontrar el literal de la cadena.

Y hablando de variables en general, un último detalle:

En un programa, las variables A, A=, A!, A%, A$ son consideradas diferentes por el ordenador y por lo tanto pueden usarse al mismo tiempo en un programa.

Area de la tabla de matrices

Las matrices pueden ser de los mismos tipos que las variables simples.

Podemos dimensionar los diferentes tipos de matrices para que contengan valores normales

DIM A (dimensiones)

de doble precisión

DIM A= (dimensiones)

de simple precisión

DIM A! (dimensiones)

valores enteros

DIM A% (dimensiones)

o cadenas de carácteres

DIM A$ (dimensiones)

Al igual que en el caso de las variables, los diferentes tipos de matrices pueden usarse al mismo tiempo en un programa, ya que son consideradas diferentes por el ordenador.

Tabla de funciones y sus códigos

Todos los códigos de esta tabla van precedidos por 255.

ABS	134	ASC	149	ATN	143
BIN$	157	CDBL	160	CHR$	167
CINT	158	COS	140	CSNG	138
CVD	170	CVI	168	CVS	210
DSKF	166	EOF	171	EXP	238
FIX	161	FPOS	167	FRE	227
HEX$	155	INP	144	INT	219
LEFT$	129	LEN	146	LOC	217
LOF	173	LOG	138	LPOS	228
MID$	131	MKD$	176	MKI$	198
MKS$	175	OCT$	154	PAD	250
PDL	164	PEEK	151	POS	242
RIGHT$	130	RND	136	SGN	245
SIN	137	SPACE$	153	SQR	135
STICK	162	STRIG	163	STR$	147
TAN	141	VAL	148	VPEEK	152

Figura 3

El descriptor de la matriz en la tabla de matrices puede tener un número variable de *bytes* y su longitud se puede hallar a partir de la información de los *bytes* cuarto y quinto del mismo.

Pero empecemos por el primero de los*bytes* que describen a una matriz determinada dentro de la tabla de matrices y que contiene un número que indica la naturaleza de la matriz.

Si el contenido del primer *byte* es un 8, significa que la matriz es de doble precisión, si contiene u 4 significa simple precisión, un 2 denotará a una matriz con valores enteros, y finalmente un 3 contendrán los descriptores de las matrices alfanuméricas.

Los dos *bytes* siguientes guardan el nombre que se le da a la matriz.

Los *bytes* cuarto y quinto contiene el número de *bytes* que hay a partir del sexto hasta el final del descriptor.

El sexto *byte* indica el número de dimensiones que tiene la matriz.

Por cada dimensión indicada por el sexto *byte*, hay a continuación dos *bytes* que indican el tamaño de la dimensión.

Finalmente hay ordenados un determinado número de *bytes* para cada elemento de la matriz.

Dicho número de *bytes* varía según el tipo de variables y coincide con el expresado en el primer *byte* del descriptor.

El espacio de trabajo

Esta es la zona a costa de la cual crecen todas las otras zonas de variables de la memoria *RAM*.

Su tamaño, que se puede hallar mediante sus direcciones límite o bien mediante el comando *PRINT FRE* (0), es máximo al encender el ordenador y va decreciendo a medida que crece el programa *BASIC*.

Las zonas situadas en direcciones por debajo del espacio de trabajo y que ya hemos visto, crecen hacia arriba, o sea, la dirección de inicio de estas zonas, que varía por motivos ajenos a las mismas, tiene un valor más bajo que su dirección final, la cual crece por necesidades de la propia zona.

A continuación del espacio de trabajo están las zonas que crecen hacia las direcciones bajas de la memoria, esto quiere decir que su dirección límite inicial, la que varía por necesidades de las otras zonas, tiene un valor más alto que su dirección límite final, que varía por necesidades de la propia zona.

La última zona de la *RAM* es la zona de variables del sistema, y es



Foto 1: *El conjunto formado por el Superexpander y monitor, además de mejorar ostensiblemente el aspecto del ordenador, lo potencia hasta cotas insospechadas.*

una zona que no varía de longitud a lo largo del programa.

La pila de BASIC

Esta es la primera de las zonas que hemos mencionado creciente hacia las direcciones inferiores de la memoria.

La pila es un método de almacenamiento de datos en memoria usado en programación.

En este caso es una pila con estructura *LIFO,* abreviatura inglesa de LAST IN FIRST OUT que significa «que el último dato que entra es el primero que sale».

De esta forma, fijando el inicio de la pila en una dirección de memoria, se van introduciendo datos en las direcciones correlativamente inferiores al mismo tiempo que el llamado puntero de la pila va decreciendo para ir señalando o apuntando a la dirección donde se encuentra el último dato introducido.

Cuando se reclame un dato de la pila, se leerá en la dirección que indica el puntero, a la vez que éste se incrementa para señalar el dato anterior.

El uso de la pila de *BASIC* es normalmente transparente para el programador, ya que es el programa traductor de la *ROM* el que la usa para anotar la dirección de retorno en comandos *GOSUB* y en los bucles *FOR-NEXT.*

No obstante, el programador ha de tener cuidado en sus programas para no dejar en la pila datos innecesarios que puedan depauperarla.

Cada sentencia *GOSUB* coloca en la pila una dirección de retorno, que ocupa 7 *bytes* de la misma.

Este dato permanece en la pila hasta que se procesa un comando *RETURN.*

Si se realiza un programa en el que haya sentencias *GOSUB* que llamen a una rutina de la cual no se salga con un comando *RETURN,* nos encontraremos con que la dirección introducida por *GOSUB* en la pila se queda en ella y ocupa un espacio de la misma, haciendo una ocupación innecesaria de la memoria.

Así mismo, los bucles *FOR-NEXT* utilizan 25 *bytes* de la pila que son limpiados solamente cuando dicho bucle se ha realizado en todos sus ciclos.

Si introducimos dentro del bucle un salto a una línea fuera del mismo, habremos dejado en la pila los 25 *bytes* usados por el bucle *FOR-NEXT,* con la consiguiente depauperación de dicha pila.

Para eludir este problema debemos colocar los bucles *FOR-NEXT* dentro de sub-rutinas y asegurarnos de que las salidas de las mismas se realizan mediante sentencias *RETURN,* la cual, además de borrar de la pila los 7 *bytes* de la dirección de retorno, elimina también los 25 *bytes* usados por el bucle *FOR-NEXT.*

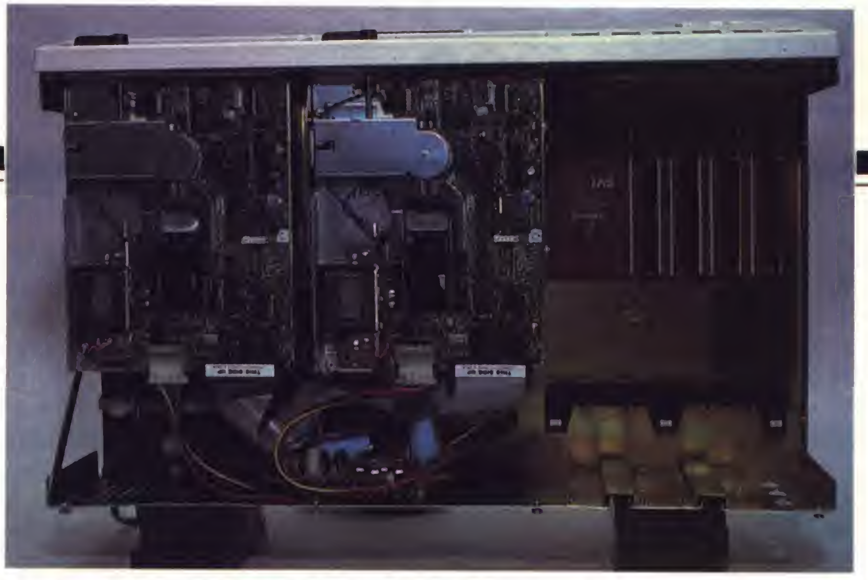


Foto 2: ***Vista interna del Superexpander, donde podemos destacar, además de dos unidades de discos de 5.25 pulgadas, buses de expansión.***

Area de las cadenas de carácteres

La segunda zona con la característica de crecimiento hacia las direcciones bajas de memoria es el área destinada a las cadenas de carácteres.

En el apartado que describe la zona de la *RAM* dedicada a la tabla de variables hemos visto que las variables alfanuméricas están descritas mediante 6 *bytes.*

Los *bytes* quinto y sexto eran respectivamente el *byte* bajo y el *byte* alto de la dirección de memoria donde se podría encontrar el literal de la cadena.

La dirección antes mencionada se hallará aplicando la fórmula:

dirección = byte bajo + 256 * byte alto

Cuando las variables están definidas literalmente en el listado del programa, la dirección antes mencionada cae dentro de la zona del programa, que es la primera que hemos visto y coincide con la dirección en que se introdujo la variables al hacer el programa.

Sin embargo, cuando en el programa se da la variable como resultado de una operación con cadenas

(Ej.: A$ = B$ + C$)

o de la transformación de un valor numérico en alfanumérico,

(Ej.: A$ = CHR$(A))

el literal de la variable resultante se

almacena en el área de memoria que estamos viendo, el área de cadenas; y la dirección señalada en los *bytes* quinto y sexto del descriptor correspondiente en la tabla de variables, será una dirección perteneciente al área de cadenas.

El tamaño del área de cadenas no varía dinámicamente con las necesidades del traductor del *BASIC*.

Al encender el ordenador se asignan 200 *bytes* iniciales para el área de cadenas.

Este valor puede ser fácilmente alterado posteriormente mediante la instrucción *CLEAR*.

(Ej.: CLEAR 3000 aumentará a 3000 octetos el área de cadenas)

El puntero de cadenas (ver direcciones límite en la figura 2), marca en todo momento la división entre el espacio Ide cadenas ocupado y el libre. Esto quiere decir que todas las direcciones por encima del puntero y hasta el inicio del área de cadenas estarán ocupadas, y todas las direcciones por debajo, incluido el puntero, hasta el final del espacio de cadenas son las que quedan libres.

El ordenador señalará cuándo el área de cadenas se queda pequeña mostrando el mensaje "out of string space».

Cuando una cadena varía su contenido, el ordenador la considera como nueva a la hora de colocarla en el espacio de cadenas, por lo tanto, la situará en la dirección que señala el puntero de cadenas variando el valor de éste hasta la dirección correspondiente, y variando así mismo el descriptor de la cadena en la tabla de variables para acoger a los nuevos datos.

Las cadenas fuente de la modificación no son eliminadas inmediatamente; cuando el área de cadenas está llena, el ordenador realiza automáticamente lo que los ingleses llaman

«garbache collection»

(algo así como «recogida de basuras»), que consiste en borrar todas las cadenas no utilizadas y ordenar las que quedan en uso a partir del comienzo del área de cadenas, lo cual puede llevar un buen rato si el área de cadenas es grande.

Al área de buffers

Un buffer o memoria intermedia es una porción de memoria que usa el ordenador para formatear y controlar los datos de E/S de varios periféricos.

En nuestro caso, la cantidad de memoria reservada es de 268 *bytes* para cada buffer utilizado.

Al encender el ordenador se asigna espacio para dos *buffers* o archivos de *E/S* llamados *FILE=0* y *FILE=1*.

El *buffer=0* está localizado en el comienzo del área de *buffers* y no debe ser eliminado pues allí realiza el ordenador varias de sus operaciones.

El *buffer=1* situado a continuación sí puede ser eliminado fácilmente mediante el comando *MAXFILES* que también se puede usar para reservar espacio para más archivos.

MAXFILES 0 eliminará el *buffer=1* haciendo que el área de cadenas y la pila de *BASIC* aumenten su dirección de localización en 268 *bytes*, dejando libre así más memoria en la zona denominada área de trabajo.

En cambio *MAXFILES* 4 reserva espacio para los *buffers FILE=2, FILE=3*, y *FILE=4*.

La dirección de inicio de cada *buffer* o archivo puede hallarse mediante la fórmula:

A = VARPTR (= número de archivo) + 65536

La razón de sumar 65536 es porque el ordenador usa números enteros para manejar direcciones. Los enteros en el *SVI* abarcan desde −32768 a 32767; para direcciones mayores de 32767 se usa el complemento a dos que da como resultado un número negativo.

Area para C.M. de usuario y sistema de disco

Si encendemos el ordenador sin tener conectadas ninguna unidad de disco y hallamos la dirección límite llamada «tope de la memoria *BASIC*» mediante la variable del sistema *HIMEM* (ver figura 3), nos encontraremos con que dicha dirección coincide con el inicio del área llamada de variables del sistema, o lo que es lo mismo, que el área que estamos tratando no existe.

Esto se debe a que este área no se usa a menos que tengamos conectada una unidad de disco y hayamos inicializado el ordenador con un disco que contenga el lenguaje extendido para disco, lenguaje que queda almacenado en este área.

También se creará este área cuando nosotros reservemos direcciones de memoria en ella para colocar allí nuestros propios programas en *C.M.*

Para reservar un espacio encima del tope de memoria usaremos el comando *CLEAR*.

(Ej.: CLEAR 200,&HD546)

Ya vimos antes cómo este co-

mando servía para reservar espacio en la zona de cadenas.

Mediante un segundo operando del comando *CLEAR* podemos variar la posición de la *RAMTOP* o dirección tope de la memoria *BASIC*.

(Ej.: línea 10 del programa 1)

El nombre de RAMTOP se debe a que por encima de esta dirección las rutinas en código máquina introducidas están a salvo de poder ser alteradas por fluctuaciones del programa basic.

Hay que tener en cuenta al poner nuestras rutinas, que por encima de la dirección *F500 hex* se encuentran las variables del sistema que no pueden ser cambiadas sin correr el riesgo de producir alteraciones graves para el sistema (corregibles solamente apagando y volviendo a encenderse el ordenador), que pueden producir la pérdida de varias horas de trabajo.

Asimismo, si trabajamos con disco, antes de alterar la *RAMTOP* procederemos, nada más encender el ordenador, a asegurarnos de la posición en que se encuentra situada. Ya que entre dicha *RAMTOP* y la dirección *F500 hex* se sitúa el lenguaje *BASIC* extendido para disco, introducido allí desde el propio disco, y que no debemos alterar tampoco.

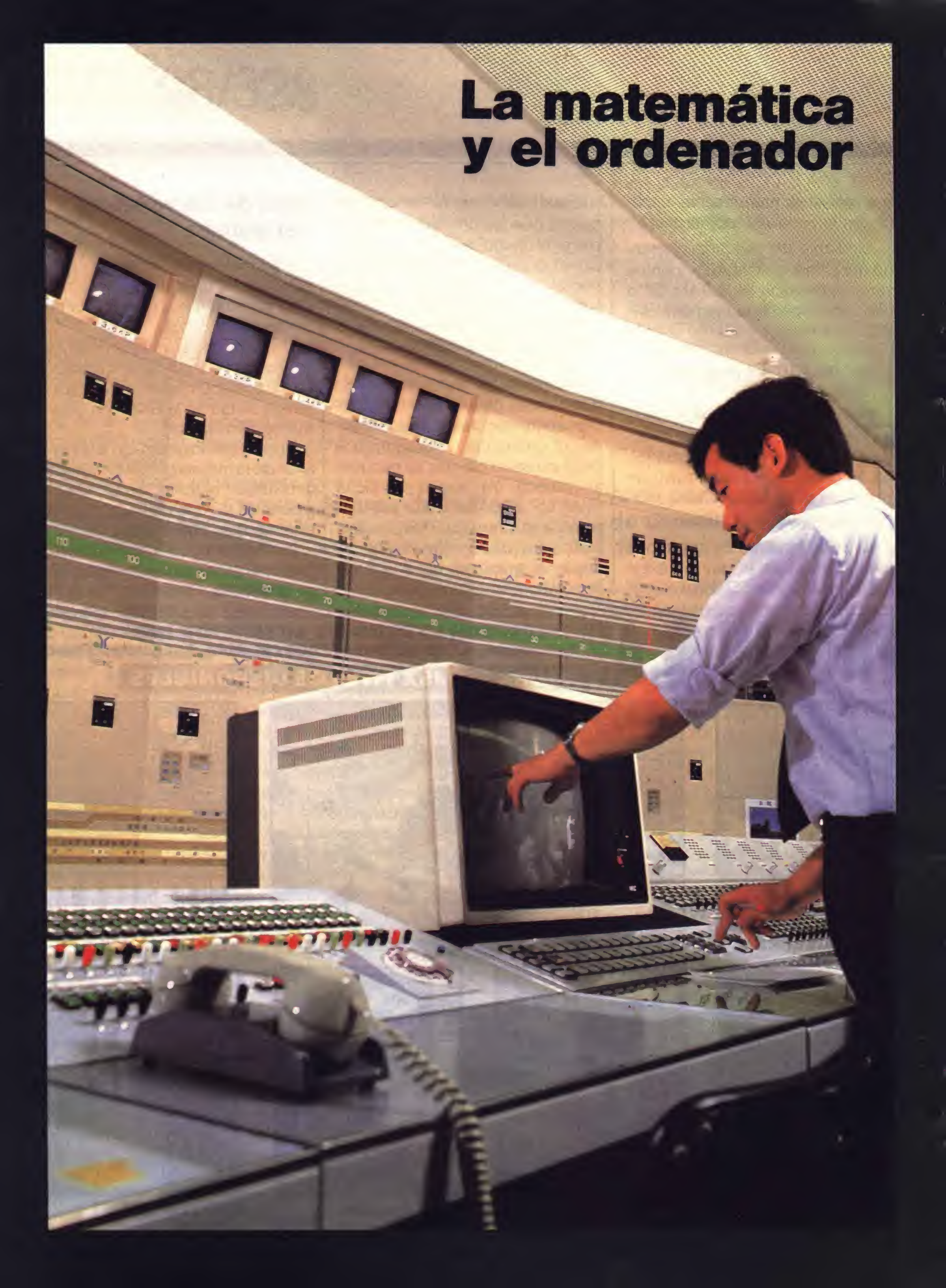
Area de las variables del sistema

Este área es usada por el ordenador como un cuaderno de notas para guardar datos que más tarde necesitará recordar.

Dichos datos se pueden referir a la posición del cursor en pantalla, las direcciones límite de las diferentes zonas de la *RAM*, los colores de pantalla, etc.

En el capítulo dedicado a las rutinas de la memoria *ROM* veremos con detalle algunas de las direcciones de la zona de variables del sistema más importantes.

En el siguiente capítulo analizaremos el procesador de vídeo, no os lo perdáis.



La matemática y el ordenador

Sistemas de ecuaciones (y III)

A lo largo de los dos últimos números de MSX MAGAZINE hemos expuesto tres métodos para la resolución de sistemas de ecuaciones lineales. Hemos hablado de sus ventajas e inconvenientes y comentado la utilidad que representa el disponer de uno o varios métodos potentes para resolver tales sistemas. Cada uno sabrá, posteriormente, aplicar estos métodos.

El último método que vamos a conocer es el Método de *Sobreiteración*. Este método es muy parecido al de *Gauss-Seidel*, pero (según algunos autores) puede llegar a ser 10 veces más rápido que éste.

Al disponer de varios métodos, podremos resolver el mismo sistema con varios o con todos ellos, comprobando, de esta forma, los resultados. Al mismo tiempo podemos hacernos una idea del orden de exactitud del resultado.

Al final hemos listado un programa que comprende los 4 métodos explicados. Esto disminuye algo la capacidad de memoria del ordenador, pero no es nada grave y nos permite utilizar varios métodos sin necesidad de tener que cargar varios programas y recuperar los datos de cinta.

Método de Sobreiteración

Sea $x^k = (x_1^k, x_2^k, \ldots x_n^k)$ el conjunto de soluciones intermedias a que hemos llegado en la iteración k aplicando el método de *Gauss-Seidel* (ver el número anterior de MSX MAGAZINE). Se puede decir que la diferencia

$$\delta x^k = x^k - x^{k-1}$$

es una estimación de

$$\delta xz = x - x^{k-1}$$



donde x es el valor exacto. De esta forma vemos que

$$x^k = x^{k-1} + \delta x^k$$

y, en general δx será, en valor absoluto, mayor que δx^k:

$$|\delta x| > |\delta x^k|$$

Parece, por tanto, comprensible, ya que no podemos conocer δx, tomar en lugar de δx^k un número mayor que él en valor absoluto. Así podemos escribir:

$$x^k = x^{k-1} + f \cdot \delta x^k$$

siendo $f > 1$.

Se suele tomar un valor de f comprendido entre 1 y 2. (Se puede encontrar un buen valor de f a partir de los autovalores de la matriz de los coeficientes del sistema de ecuaciones; pero aquí no lo haremos). Particularmente hemos comprobado que en la mayor parte de los casos f está comprendido entre 0.9 y 1.2 en sistemas de diagonal dominante.

De esta forma para hallar los valores x de la iteración k–ésima, podemos proceder de la siguiente forma:

a) Calculamos el valor intermedio:

$$y_i^k = b_i - (\sum a_{ij} x_j^k + \sum a_{ij} x_j^{k-1}) \qquad a_{ii}$$

b) Calculamos el valor x_i^k en la iteración k por:

$$x_i^k = x_i^{k-1} + F\,(y_i^k - x_i^{k-1})$$

Recordemos que a_{ij} son los elementos de la matriz de los coeficientes y b_i es el término independiente de la ecuación i.

Si resolvemos por este procedimiento el sistema:

$$\begin{aligned} 3x + 2y + z &= 10 \\ 2x + 3y + z &= 11 \\ x + 2y + 3z &= 14 \end{aligned}$$

para un valor de f=1.14 se llega al resultado en la iteración 23. El método de *Gauss-Seidel* (f=1) emplea 39 iteraciones y si f=1.2, se necesitan 25 iteraciones para llegar a la solución.

El diagrama 1 esquematiza la rutina de trabajo de este método.

Como decíamos al principio, el programa que acompaña a este artículo incluye los cuatro métodos de que hemos hablado. El primer *MENU* permite elegir entre ellos. Con el segundo *MENU* se puede seleccionar la procedencia de los datos. Para introducir los datos a través del teclado pulsaremos la opción 1. Estos datos pueden ser grabados en cinta una vez introducidos pulsando la tecla de función F4 (al pulsar F4, el programa nos pedirá el nombre del archivo. Si se omite éste, el archivo toma el nombre *"DATOS"*). La opción 2 nos permite tomar los datos de cinta magnetofónica (igualmente si se omite el nombre del archivo, éste pasa a denominarse *"DATOS"*). Si, una vez calculadas las raíces por un procedimiento, queremos probar con otro (utilizando los mismos datos), pulsaremos la opción 0 del segundo menú.

Una vez leídos los datos del teclado o de cinta, estos se listarán automáticamente para su comprobación (con ayuda del cursor). Si hay que modificar algún dato, pulsaremos la tecla de función F3. En caso contrario, pulsaremos la barra espaciadora, y comenzará la ejecución.

Se han introducido dos nuevas posibilidades: Podemos dar un vector inicial de soluciones $x^0 = (x_1^0, x_2^0, ... x_0^n)$, o bien pasar de ello, con lo que automáticamente se toma un vector inicial $x^0 = (1,1,...1)$. También se tiene opción a listar o no todos los resultados intermedios. (Se puede detener el listado temporalmente pulsando la barra espaciadora. Pulsándola de nuevo, el programa continúa).

Por lo demás, pulsando F1 se actúa el motor de la grabadora, F2 nos permite en cualquier momento listar los elementos de la matriz ampliada y F5 presenta una página de ayuda.

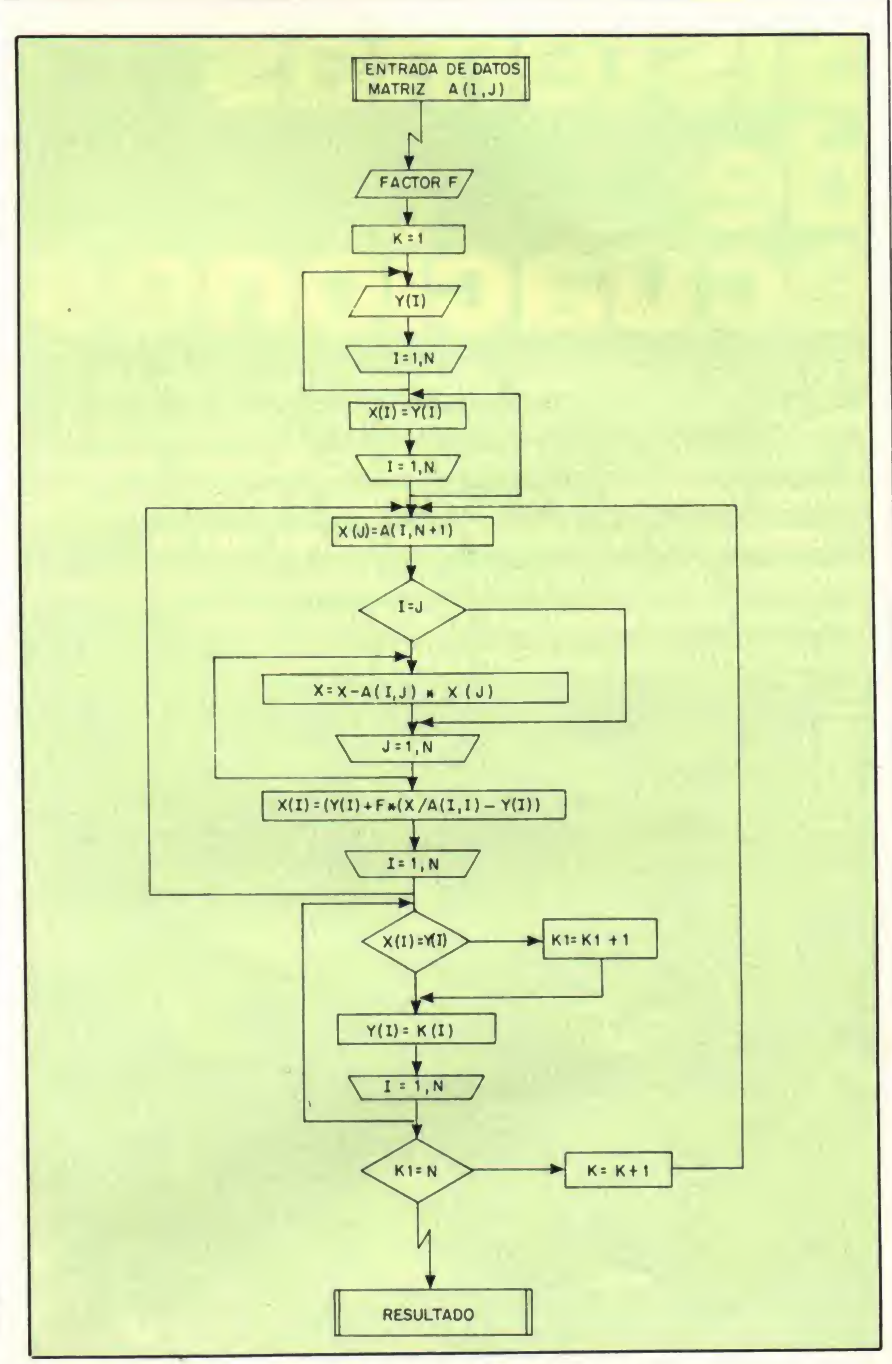
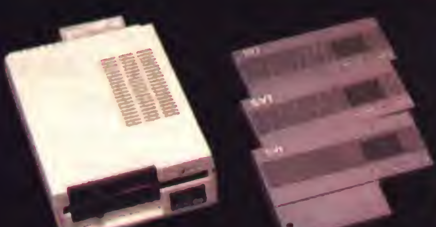


Diagrama 1. Método de Sobreiteración.

J. Antonio Feberero

```
10 'SISTEMAS DE ECUACIONES
20 'Versión 6.12.03.86 - 5148 Bytes
30 'Juan Antonio Feberero Castejón
40 '
50 CLEAR 200,60000!:SCREEN 0,,0:WIDTH 39
   :KEY OFF
60 DEFINT I-N:DEFSTR W
70 ON ERROR GOTO 15040
100 '
110 'INTRO DATOS
120 '===========
130 '
140 CLS:LOCATE 4,5:PRINT "SISTEMAS DE EC
   UACIONES LINEALES",TAB(4);STRING$(3
   1,195)
150 LOCATE 5,8:PRINT "<1> Método de Gaus
   s-Jordan"
160 LOCATE 5,10:PRINT "<2> Método de Jac
   obi"
170 LOCATE 5,12:PRINT "<3> Método de Gau
   ss-Seidel"
180 LOCATE 5,14:PRINT "<4> Método de Sob
   reiteración"
190 W=INKEY$:IF W="" THEN 190
200 W%=INSTR("1234",W)
210 IF W%=0 THEN 190
220 ON KEY GOSUB 10000,11000,12040,13040
   ,14040
230 KEY(1)ON:KEY(5)ON
240 CLS:LOCATE 3,8:PRINT "<0> Datos ya e
   n memoria"
250 LOCATE 3,10:PRINT "<1> Datos procede
   ntes de teclado"
260 LOCATE 3,12:PRINT "<2> Datos procede
   ntes de cinta
270 W=INKEY$:IF W="" THEN 270
280 IF INSTR("012",W)=0 THEN 270
290 IF VAL(W)=0 THEN 840
300 CLS:IF VAL(W)=2 THEN 640
400 '
410 'Datos de teclado
420 '----------------
430 '
440 INPUT "Nº de ecuaciones (Máx. 48)";N
450 DIM A(N,N+1),X(N),Y(N)
460 FOR I=1 TO N
470 FOR J=1 TO N
480 PRINT USING "A(##;##):";I;J;:INPUT A
   (I,J)
490 NEXT J
500 PRINT USING "B(##)    :";I;:INPUT A(I
   ,N+1)
510 NEXT I
520 GOTO 840
600 '
610 'Datos de cinta
620 '--------------
630 '
640 PRINT "Preparar grabadora","Pulsar [
   RETURN]
650 WARCH="DATOS"
660 PRINT:PRINT "(Todos los datos almace
   nados como cade-nas de caracteres)"
   ,,,"Nombre del archivo [RETURN]";:I
   NPUT WARCH
670 OPEN WARCH FOR INPUT AS#1
680 PRINT:PRINT "Archivo ";WARCH;": enco
   ntrado"
690 LINE INPUT#1,WN
700 N=VAL(WN)
710 DIM A(N,N+1),X(N),Y(N)
720 FOR I=1 TO N
730 FOR J=1 TO N+1
740 IF EOF(1) THEN 790
750 LINE INPUT#1,W
760 A(I,J)=VAL(W)
770 NEXT J
780 NEXT I
790 CLOSE #1
800 '
810 'LIST. DATOS
820 '===========
830 '
840 CLS
850 KEY(2)ON:KEY(3)ON:KEY(4)ON
860 FOR I=1 TO N
870 FOR J=1 TO N+1
880 IF J=N+1 THEN PRINT USING "B(##)    =
   &":I:STR$(A(I,N+1)):GOTO 920
890 PRINT USING "A(##:##)=&":I;J;STR$(A(
   I,J))
900 IF CSRLIN=20 THEN 920
910 GOTO 980
920 W=INKEY$:IF W="" THEN 920
930 IF W=CHR$(30) AND I*J<N*N+N THEN 970
940 IF W=CHR$(31) AND J<N+2 THEN J=J-CSR
   LIN-20:IF J<-1 THEN I=I-2:GOTO 990
   ELSE 970
950 IF W=CHR$(32) THEN 1010
960 GOTO 920
```









- Capacidad de Memoria de 80 K RAM y 32 K ROM, con el intérprete de BASIC (MICROSOFT) incorporado.
- Teclado de diseño ergonómico, con teclas numéricas independientes.
- Gráficos de alta resolución: red de 296 por 152 puntos, con 16 colores disponibles.
- Conexiones para TV, monitor, unidad de disco de 320 K. (incluido sistema operativo CP/M), impresora, cassettes y cartuchos MSX.
- Conector para dos Joysticks.
- Ordenador MSX.

Aquí y ahora, con toda su potencia







- Sistemas operativos: CP/M, MSX-BASIC y MSX-DOS.
- Con una memoria de 80 K RAM.
- Unidad de disco de 3′ 5″ integrada en la consola del teclado.
- Dos puertas de conexión: RS232-C y Paralelo Centronics.
- Salidas directas a televisor y monitor.
- Admite directamente la conexión de una segunda unidad de disco, sin necesidad de interface o cartucho de ampliación de memoria.
- Posibilidad de operar en 40 ó 80 columnas sin necesidad de cartucho.
- Transporte cómodo: asa incorporada y maletín de transporte.

SVI

SPECTRAVIDEO

Desde ahora SVI-Spectravídeo está aquí.

Con toda la potencia de su nombre y su organización mundial, con toda su tecnología de futuro.

SVI-Spectravídeo marca el comienzo de una nueva era.

Muchas cosas se van a quedar en el pasado.

Y, probablemente, nada será igual a partir de ahora.

Con SVI-Spectravídeo, la vida diaria puede hacerse más sencilla, con más posibilidades, más divertida y excitante, tanto en casa como en el trabajo.

Porque, desde ahora, y para el futuro, puedes contar con SVI-Spectravídeo, España.

Aquí, con toda su potencia.

Y, siempre, muy cerca de ti: en 1.500 puntos de venta.

SVI-Spectravídeo, España, significa la más avanzada tecnología, mantenida por el servicio técnico más exigente y eficaz, en:

Ordenadores. Periféricos y accesorios. Joysticks. Juguetes electrónicos. Sonido Hi-Fi.



```
970 CLS
980 NEXT J
990 CLS:IF I<0 THEN I=0
1000 NEXT I
1010 CLS
1020 KEY(3)OFF:KEY(4)OFF
1030 IF W% >1 THEN 2040
1500 '
1510 'GAUSS-JORDAN
1520 '============
1530 '
1540 FOR K=1 TO N-1
1550 PIVOTE=ABS(A(K,K)):FILA=K
1560 FOR J=K+1 TO N
1570 IF ABS(A(J,K))>PIVOTE THEN PIVOTE=A
     BS(A(J,K)):FILA=J
1580 NEXT J
1590 IF FILA>K THEN FOR J=K TO N+1:SWAP
     A(K,J),A(FILA,J):NEXT
1600 IF PIVOTE=0 THEN CLS:LOCATE 10,11:P
     RINT "SISTEMA NO RESOLUBLE":END
1610 FOR I=K+1 TO N
1620 A=A(I,K)/A(K,K)
1630 FOR J=1 TO N+1
1640 A(I,J)=A(I,J)-A*A(K,J)
1650 NEXT J
1660 NEXT I
1670 NEXT K
1680 X(N)=A(N,N+1)/A(N,N)
1690 FOR I=N-1 TO 1 STEP-1
1700 X(I)=A(I,N+1)
1710 FOR J=I+1 TO N
1720 X(I)=X(I)-A(I,J)*X(J)
1730 NEXT J
1740 X(I)=X(I)/A(I,I)
1750 NEXT I
1760 GOTO 4040
2000 '
2010 'ITERATIVOS
2020 '==========
2030 '
2040 INPUT "Desea dar valores iniciales:
      S/N";W
2050 IF W="N" OR W="n" THEN 2110
2060 FOR I=1 TO N
2070 PRINT USING "X(##)";I;
2080 INPUT Y(I)
2090 NEXT I
2100 GOTO 2130
2110 FOR I=1 TO N:Y(I)=1:NEXT
2120 K=0
2130 INPUT "Desea ver iteraciones: S/N";
     WIT$
2140 ON W% GOTO ,2240,2540,2540
2200 '
2210 'JACOBI
2220 '======
2230 '
2240 FOR I=1 TO N
2250 X=A(I,N+1)
2260 FOR J=1 TO N
2270 IF J=I THEN 2290
2280 X=X-A(I,J)*Y(J)
2290 NEXT J
2300 X(I)=X/A(I,I)
2310 NEXT I
2320 GOSUB 3040
2330 GOTO 2240
2500 '
2510 'GAUSS-SEIDEL Y SOBREITERACION
2520 '============================
2530 '
2540 F=1:IF W%=3 THEN 2560
2550 INPUT"Factor F";F
2560 FOR I=1 TO N
2570 X(I)=Y(I)
2580 NEXT I
2590 FOR I=1 TO N
2600 X=A(I,N+1)
2610 FOR J=1 TO N
2620 IF I=J THEN 2640
2630 X=X-A(I,J)*X(J)
2640 NEXT J
2650 X(I)=Y(I)+F*(X/A(I,I)-Y(I))
2660 NEXT I
2670 GOSUB 3040
2680 GOTO 2590
3000 '
3010 'Presentación de iteraciones
3020 '----------------------------
3030 '
3040 K1=0
3050 FOR I=1 TO N
3060 IF X(I)=Y(I) THEN K1=K1+1
3070 Y(I)=X(I)
3080 NEXT I
3090 IF K1=N THEN RETURN 4040 ELSE K=K+1
```

```
3100 IF WIT$="N" OR WIT$="n" THEN BEEP:R
     ETURN
3110 PRINT:PRINT "Iteración nº":k
3120 FOR I=1 TO N
3130 PRINT USING "x(##)=&";I;STR$(Y(I))
3140 NEXT I
3150 IF INKEY$="" THEN RETURN
3160 IF INKEY$="" THEN 3160 ELSE RETURN
4000 '
4010 'Resultados
4020 '----------
4030 '
4040 PRINT:PRINT:PRINT "RESULTADOS:"
4050 FOR I=1 TO N
4060 PRINT USING "x(##)=&";I;STR$(X(I))
4070 IF CSRLIN=20 OR I=N THEN 4090
4080 GOTO 4150
4090 W=INKEY$:IF W="" THEN 4090
4100 IF W=CHR$(30) AND I<N THEN 4140
4110 IF W=CHR$(31) AND I>20 THEN I=I-CSR
     LIN-20:GOTO 4140
4120 IF W=CHR$(32) THEN 140
4130 GOTO 4090
4140 CLS
4150 NEXT I
10000 MOTOR:RETURN
11000 RETURN 840
12000 '
12010 'Modifica
12020 '--------
12030 '
12040 CLS
12050 INPUT "Dato a modificar: FILA";I
12060 INPUT "COLUMNA (0 para término ind
      epend.";J
12070 IF J=0 THEN J=N+1
12080 INPUT "Nuevo valor";A(I,J)
12090 RETURN 840
13000 '
13010 'GRABA CINTA
13020 '===========
13030 '
13040 CLS:PRINT "Preparar grabadora [RET
      URN]"
13050 WARCH="DATOS"
13060 PRINT:PRINT "(Todos los datos se a
      lmacenan como ca-denas de caractere
      s)",,,"Nombre del archivo [RETURN]"
      ;:INPUT WARCH
13070 OPEN WARCH FOR OUTPUT AS#1
13080 PRINT#1,STR$(N)
13090 FOR I=1 TO N
13100 FOR J=1 TO N+1
13110 PRINT#1,STR$(A(I,J))
13120 NEXT J
13130 NEXT I
13140 CLOSE #1
13150 RETURN 840
14000 '
14010 'Help
14020 '----
14030 '
14040 XP=POS(0):YP=CSRLIN
14050 FOR IHELP=1 TO 960:POKE 60000!+IHE
      LP,VPEEK(IHELP):NEXT
14060 CLS
14070 PRINT "HELP",,"----"
14080 PRINT:PRINT "F1:Motor",,"F2:List.
      datos","F3:Modific. datos","F4:Grab
      ar datos en cinta","F5:Help
14090 PRINT:PRINT:PRINT "LIS. DATOS",,ST
      RING$(10,195)
14100 PRINT "Cursor:Página arriba/abajo"
      ,"Barra espaciadora:Sale de List. C
      omien-za proceso. Vuelve a List.",,
      ,,"En List. están activadas F1 a F5
      ","En otro caso están activadas F1,
       F2 Y  F5
14110 LOCATE 12,23:PRINT "Pulsa una tecl
      a";
14120 IF INKEY$="" THEN 14120
14130 CLS
14140 FOR IHELP=1 TO 960:VPOKE IHELP,PEE
      K(60000!+IHELP):NEXT
14150 LOCATE XP,YP
14160 RETURN
15000 '
15010 'Errores
15020 '-------
15030 '
15040 IF ERR=11 AND ERL=1680 OR ERL=1620
       THEN 15070
15050 IF ERR=10 THEN ERASE A,X,Y:IF ERL=
      450 THEN 450 ELSE 710
15060 CLS:LOCATE 7,12:PRINT "ERROR ";ERR
      ;" EN LINEA ";ERL:STOP
15070 CLS:LOCATE 9,11:PRINT "SISTEMA NO
      RESOLUBLE.":STOP
```

BASIC MSX puede manejar tantos datos numéricos (variables numéricas) como conjuntos de letras y signos gráficos o cadenas de caracteres (variables alfanuméricas).



SIC

Variables numéricas y alfanuméricas

Las VARIABLES NUMERICAS pueden ser de tres tipos:

VARIABLES ENTERAS: Serán variables a las que se asignará un número entero. Este número puede variar de –32768 a +32767. Si escribimos la variable **N** (entera) seguida del símbolo %, es decir: **N%,** el ordenador interpretará que se trata de una variable entera y le asignará en memoria un espacio de 2 *bytes* u octetos.

VARIABLES REALES: Si una variable numérica es real, es decir, tiene una parte entera y otra decimal, o bien excede de los límites de las variables enteras (es decir: es menor que –32768 o mayor que +32767), debemos utilizar una VARIABLE REAL. Hay dos tipos de variables reales:

Variables de simple precisión: Son variables rales con un número de cifras igual o menor que 6. Para que el ordenador las interprete, escribiremos el símbolo ! después de la letra o letras que identifican la variable. Por ejemplo **A!** y **HOLA!** son variables de simple precisión. El ordenador asigna automáticamente una memoria de 4 octetos a estas variables.

Variables de doble precisión: Son, como las anteriores, variables reales; pero éstas tienen 14 cifras significativas, lo que permite trabajar con mayor precisión que con variables de simple precisión (pero también más lento). Al conectar el ordenador, todas las variables que se utilicen son de doble precisión. Si a la variable A no se le añade uno de los símbolos %, o $ (que veremos ahora), se trata de una variable de doble precisión. No obstante, también se puede añadir el símbolo # al nombre de la variable para identificarla como de doble precisión, así: A#, HOLA#. El ordenador asigna a estas variables una memoria de 8 octetos.

VARIABLES ALFANUMERICAS: Podemos asignar a una variable alfanumérica cualquier combinación de caracteres (encerrados entre comillas), hasta un máximo de 255. Para identificar una variable alfanumérica basta añadir el símbolo $ al nombre de la variable; por ejemplo A$. Una variable alfanumérica ocupa una memoria de 3 octetos más un octeto (del área reservada para cadenas de caracteres) por cada caracter.

Los siguientes son ejemplos de variables:

ENTERAS : A%= 123 „ NUMERO%=–25429
DE SIMPLE PRECISION: A!= 123456 „ SIMPLE!=–2.345
DE DOBLE PRECISION : A#= 1234567890 „ DOBLE#=–3.141E25
ALFANUMERICAS : A$= "ALFANUMERICA" „ ALFA$= "HOLA, QUE TAL?"

El editor de pantalla es un conjunto de códigos que nos permite escribir un programa y modificarlo.

Notas

1. Los números reales multiplicados por potencias de 10 se escriben añadiendo al número la letra E (exponente) o la letra D (exponente en números de doble precisión); por ejemplo, $1.234 \cdot 10^{-25}$ se escribe: 1.234E–25.

El exponente de 10 debe ser un número entero comprendido entre –64 y +63.

2. Las variables A%, A!, A# o A,

y A$ son distintas para *BASIC* MSX; es decir se pueden utillizar todas ellas simultáneamente con valores distintos sin temor a error.

3. Si queremos que todas las variables que empiecen por las letras I, J, K, L, M, N, por ejemplo, sean de tipos entero, no es necesario escribir el símbolo de tipo (%) detrás de cada una de ellas si previamente (preferiblemente al principio del programa) se hace una definición de tipo:

```
10 DEFINT I, J, K, L, M, N
```

o bien:

```
10 DEFINT I–N
```

Si posteriomente se quiere utilizar una variable que empiece por una de estas letras con otro tipo, basta con añadir al nombre de la variable el símbolo del tipo. Por ejemplo, si queremos utilizar la variable NOMBRE como de cadena de caracteres, se escribirá NOMBRE$ para esta variable.

Igualmente, se pueden definir conjuntos de variables, todas ellas identificadas por la letra inicial, de simple y doble precisión y alfanuméricas, mediante las definiciones de tipo *DEFSNG, DEFCBL* y *DEFSTR,* respectivamente (STR viene del inglés STRING = cadena de caracteres). Por ejemplo:

```
10 DEFINT I–N:DEFSNG T,H:-
   DEFDBL E–G:DEFSTR A
```

IF...THEN...ELSE

Vamos a preparar otro programa. Esta vez queremos calcular las raíces (soluciones) de una ecuación de segundo grado.

Una ecuación de 2 grado tiene la forma:

$$ax^2 + bx + c = 0$$

donde a, b y c son valores conocidos y x es desconocido (incógnita).

Toda ecuación de 2.º grado tiene 2 soluciones, y estas son:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Pero sabemos que no es posible hallar la raíz cuadrada de un número negativo. Por tanto, si $b - 4ac$ es negativo, no existirán valores (reales) de x. Es decir, nuestra ecuación no tendrá solución.

Así tendremos primero que ver si:

$$d = b^2 - 4ac$$

es positivo. Si lo es, la ecuación tendrá solución, y procederemos a calcular x_1 y x_2.

Los signos de puntuación también cumplen una función determinada, según el caso.

Del programa de la figura 1 ya conocemos casi todo, pero no entendemos lo que quiere decir la línea 100. Se trata de una SENTENCIA DE CONTROL CONDICIONAL que debemos leer así:

"SI D ES MENOR QUE CERO ENTONCES IR A LA LINEA 160; EN CASO CONTRARIO HACER D IGUAL A LA RAIZ CUADRADA DE D".

De esta forma, vemos que cada vez que queramos elegir entre dos opciones, basta con utilizar la sentencia:

IF (Expresión lógica) *THEN* (Expresión 1) *ELSE* (Expresión 2)

Así, si la Expresión lógica es cierta, se ejecutará lo que diga la Expresión 1. En caso contrario, se ejecutará lo que diga la Expresión 2. (NOTA: Expresión 1 y Expresión 2 también pueden ser sentencias de control condicional).

También se puede escribir:

IF (Expresión lógica) *THEN* (Expresión 1)

de forma que si Expresión lógica es falsa no se ejecutará Expresión 1 y el programa pasará a la línea siguiente.

La Expresión lógica es una expresión en la que se establece una o varias comparaciones entre una o varias variables, o bien se analiza el valor de una variable de tipo lógico.

Entre dos variables numéricas (o alfanuméricas) A y B, puede ocurrir:

1. que A sea igual a B: A = B
2. que A sea mayor que B: A > B
3. que A sea mayor o igual que B: A >= B
4. que A sea menor que B: A < B
5. que A sea menor o igual a B: A <= B
6. que A sea distinto de B: A <> B

A la derecha se ha escrito la expresión utilizada en *BASIC.*

Así vemos que para saber si D (calculada en la línea 90) es menor que 0, basta preguntar si:

D < 0

Lo que hacemos en la línea 100 es, por tanto preguntar si D es menor que 0. Si es así, el programa se bifurcará a la línea 160 (Raíces imaginarias), y si no (es decir, si D es mayor o igual que 0), entonces el programa hará *D=SQR(D).*

SQR(X) es una función incorporada de *BASIC* MSX que calcula la raíz cuadrada del argumento X.:

SQR(X) X

```
10 REM ECUACION DE SEGUNDO GRADO
20 REM
30 SCREEN 0,,0
40 PRINT "ECUACION DE SEGUNDO GRADO"
50 PRINT "Ax2+Bx+C=0"
60 PRINT
70 INPUT "A,B,C"; A,B,C
80 PRINT
90 D=B*B-4*A*C
100 IF D<0 THEN 160 ELSE D=SQR(D)
110 X1=(-B+D)/2/A
120 X2=(-B-D)/2/A
130 PRINT "X1=";X1
140 PRINT "X2=";X2
150 GOTO 170
160 PRINT "RAICES IMAGINARIAS"
170 PRINT
180 INPUT "DESEAS CONTINUAR";A$
190 IF A$="SI" OR A$="si" THEN 50 ELSE END
```

Figura 1: IF... THEN... ELSE...

No debe sorprendernos al haber escrito:

D=SQR(D)

(ya sabemos que, en general, D no será igual a su raíz cuadrada). En los lenguajes de programación, lo anterior no es una igualdad, sino una SENTENCIA DE ASIGNACION (como ya hemos visto). Aquí lo que hace el ordenador es escribir en el espacio de memoria en que estaba almacenada la variable D, el valor de su raíz cuadrada.

En la línea 180 introducimos desde el teclado la variable alfanumérica A$. Como el ordenador nos pregunta "DESEAS CONTINUAR?", lo lógico es contestas SI o NO. Esto es lo que el programa analiza en la línea 190:

El BASIC tiene muchas palabras que tienen un significado especial, son las palabras reservadas.

IF A$="S" OR A$="si"
THEN 50 ELSE END

Esta línea tiene una estructura muy parecida a la de la línea 100. Pero ahora nos encontramos con una expresión algo extraña:

A$="SI" OR A$="si"

Es lógico que comparemos una variable alfanumérica *(A$)* con una constante alfanumérica *("SI")* (habrás observado que las constantes alfanuméricas se escriben entre comillas). Pero ¿qué significa ese *"OR"*?

OR es un OPERADOR LOGICO. *BASIC* MSX admite 6 operadores lógicos *(NOT, AND, OR, XOR, EQV* e *IMP).* Pero hablaremos de ellos más adelante. Ahora nos basta con conocer dos: *AND* y *OR.*
Los operadores *AND* y *OR* nos van a servir para relacionar dos (o más) comparaciones. Así:

IF (Expresión lógica 1) *AND* (Expresión lógica 2) *THEN* (Expresión)

analizará las expresiones lógicas 1 y 2. Si se cumplen (son ciertas) simultáneamente, entonces se ejecutará (Expresión) Por ejemplo:

IF A<0 AND B<0 THEN
PRODUCTOAB$="POSITIVO"

Esta expresión estudia el signo de A y el de B. Si ambos son negativos, entonces el producto de A y B es positivo, y la variable alfanumérica *PRODUCTOAB$* tomará el valor "POSITIVO".

Si en lugar de *AND* escribimos *OR,* se analizan las expresiones lógicas 1 y 2. Si una de ellas es cierta, entonces se ejecutará (Expresión). Observa que ahora basta con que sea cierta una de las dos expresiones lógicas.

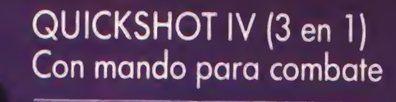
En nuestra línea 190 analizaremos si la variable *A$* es igual a "SI" o a "si", porque en la línea 180 podemos haber escrito SI con letras mayúsculas o minúsculas. Si se cumple una de las dos condiciones, entonces el programa irá a la línea 50 y nos pedirá nuevos valores de A, B y C. En caso contrario, el programa se encuentra la sentencia *END* que quiere decir FIN.



vendidos del mundo.

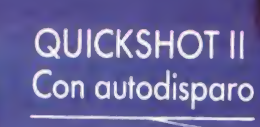


QUICKSHOT IV (3 en 1)
Con mando para combate



QUICKSHOT II MSX
Con autodisparo

QUICKSHOT II
Con autodisparo

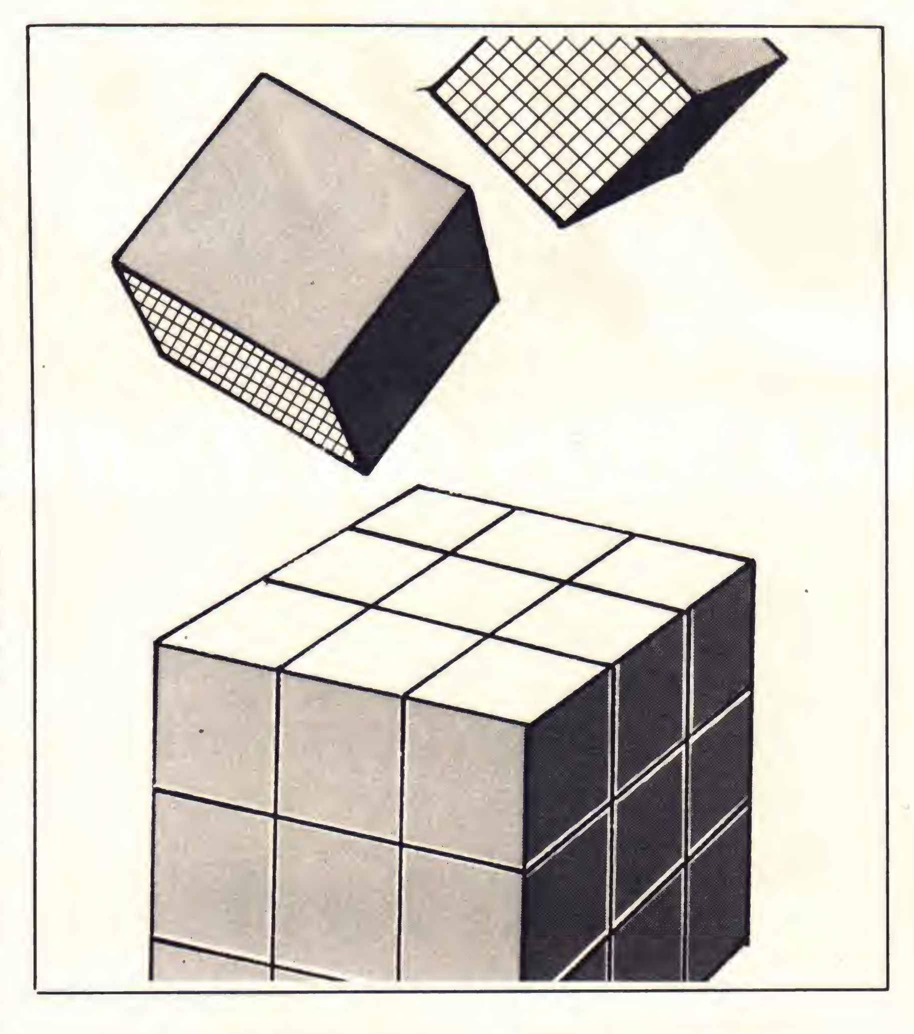


QUICKSHOT VII MSX
Portátil

Importador exclusivo SVI-España.



Puzzle



Los ratos de ocio se pueden combatir con las actividades más diversas, desde jugar una buena mano de mus hasta la realización de puzzles o rompecabezas.

Pero cuando no tengamos uno de estos últimos y estemos con el ordenador podremos jugar con este programa que hace las veces de rompecabezas. El juego consiste en ordenar las piezas que forman una figura.

Dichas piezas se encuentran desordenadas sobre la pantalla en un cuadrado de 4x4. Existe un espacio, de color rojo, hacia el cual hay que ir traslandando las piezas para ordenar la figura.

Cuando una pieza se mueve, esta intercambia su posición con la que no tiene nada. Estos movimientos se realizan con las siguientes teclas:

—*U*, desplaza hacia arriba la pieza que se encuentra debajo del espacio.

—*N*, desplaza hacia abajo la pieza que se encuentra encima del espacio.

—*H*, para mover la que está a la derecha hacia la izquierda.

—*J*, para mover la que está a la izquierda hacia la derecha.

Si se pulsa la tecla *V*, se muestra en la pantalla la figura que hemos de componer. Para volver al juego, hemos de pulsar la *V* nuevamente.

El programa está dividido en varias subrutinas que facilitan su seguimiento, así como su modificación. Si deseamos cambiar las figuras, ya que el juego está preparado para resolver dos figuras, solamente habrá que cambiar los contenidos de las instrucciones *DATA*, teniendo en cuenta que cada instrucción *DATA* corresponde a un sprite de 16x16 *pixels* ampliados.

Es un buen juego en el que podrá comprobar su habilidad y además un buen ejercicio para diseñar *sprites* mediante sentencias *DATA*.

Joaquín Fuentes Hernández
Murcia

```
10 REM PUZZLE
20 '
30 '
40 REM PROGRAMA PRINCIPAL
50 '
60 '
70 MO=0:T=0:DIM MA(4,4):COLOR 1
   5,1,4
80 GOSUB 2450:REM INSTRUCCIONES
90 CLS
100 GOTO 2090:REM PRESENTAR OPC
    ION
110 OPEN "GRP:"AS 1
120 PRESET(15,50):PRINT#1,"Un m
    omento..."
130 GOSUB 900:REM INICIALIZAR
```



LA REVISTA IMPRESCINDIBLE
PARA LOS USUARIOS DE LOS ORDENADORES PERSONALES MSX.

Una publicación mensual que ayuda a obtener el máximo partido a su ordenador.

MSX publica cada mes programas y juegos, además de reportajes sobre programación y la posibilidad de ganar premios realizando programas y otros temas siempre de gran interés.





ADEMAS, beneficiese de un **15 %** DE DESCUENTO sobre el precio real de suscripción

PRECIO NORMAL DE SUSCRIPCION	USTED SOLO PAGA	AHORRO
~~3.600~~ PTAS.	3.060 PTAS.	15 %

APROVECHE AHORA esta irrepetible oportunidad para suscribirse a **MSX**. Envie **HOY MISMO** la tarjeta adjunta a la revista, que no necesita sobre ni franqueo. Depositela en el buzón más cercano. Inmediatamente recibirá su primer ejemplar de **MSX** más el **REGALO**. Y así durante un año (12 números).



Bravo Murillo, 377
Tel. 733 79 69
28020 MADRID

```
140 A$=INKEY$:IF A$="" THEN 140
150 IF A$="U" OR A$="u" THEN 250
160 IF A$="N" OR A$="n" THEN 420
170 IF A$="H" OR A$="h" THEN 580
180 IF A$="J" OR A$="j" THEN 740
190 IF A$="V" OR A$="v" THEN GOSUB 1620
200 GOTO 140
210 '
220 '
230 '
240 '
250 REM ARRIBA
260 GOSUB 1770:REM ENCONTRAR CUADRADO ROJO
270 REM X e Y CONTIENEN LAS COORDENADAS DEL CUADRADO ROJO
280 IF Y=4 THEN GOTO 140
290 BEEP
300 X1=X*32-32:Y1=Y*32
310 FOR I=1 TO 32
320 PUT SPRITE MA(X,Y+1),(X1+10,Y1-I),15
330 PUT SPRITE MA(X,Y),(X1+10,Y1-32+I),8
340 NEXT I
350 BEEP
360 MA(X,Y)=MA(X,Y+1):MA(X,Y+1)=12
370 GOSUB 1860:REM COMPLETO ?
380 GOSUB 2020
390 GOTO 140
400 '
410 '
420 REM ABAJO
430 GOSUB 1770
440 IF Y=1 THEN GOTO 140
450 BEEP
460 X1=X*32-32:Y1=Y*32-64
470 FOR I=1 TO 32
480 PUT SPRITE MA(X,Y-1),(X1+10,Y1+I),15
490 PUT SPRITE MA(X,Y),(X1+10,Y1+32-I),8
500 NEXT I
510 BEEP
520 MA(X,Y)=MA(X,Y-1):MA(X,Y-1)=12
530 GOSUB 1860
540 GOSUB 2020
550 GOTO 140
560 '
570 '
580 REM IZQUIERDA
590 GOSUB 1770
600 IF X=4 THEN GOTO 140
610 BEEP
620 X1=X*32:Y1=Y*32-32
630 FOR I=1 TO 32
640 PUT SPRITE MA(X+1,Y),(X1-I+10,Y1),15
650 PUT SPRITE MA(X,Y),(X1-32+I+10,Y1),8
660 NEXT I
670 BEEP
680 MA(X,Y)=MA(X+1,Y):MA(X+1,Y)=12
690 GOSUB 1860
700 GOSUB 2020
710 GOTO 140
720 '
730 '
740 REM DERECHA
750 GOSUB 1770
760 IF X=1 THEN 140
770 BEEP
780 X1=X*32-64:Y1=Y*32-32
790 FOR I=1 TO 32
800 PUT SPRITE MA(X-1,Y),(X1+I+10,Y1),15
810 PUT SPRITE MA(X,Y),(X1+32-I+10,Y1),8
820 NEXT I
830 BEEP
840 MA(X,Y)=MA(X-1,Y):MA(X-1,Y)=12
850 GOSUB 1860
860 GOSUB 2020
870 GOTO 140
880 '
890 '
900 REM INICIALIZAR
910 FOR I=0 TO 15
920 C$=""
930 FOR J=0 TO 31
940 READ N
950 C$=C$+CHR$(N)
```

Philips MSX-2

el ordenador multiuso para el hogar y la oficina.

El nuevo Philips MSX-2 es un sistema completo que atraerá a un gran número de personas que usan ordenadores en casa.

Personas tales como el ejecutivo que lleva trabajo a casa, el empleado autónomo, estudiantes y secretarias.

El conjunto entre el avanzado ordenador VG 8235 y nuestro paquete de software, cubren la mayoría de las grandes áreas de aplicaciones productivas. Philips MSX-2 le ofrece un gran sistema a un precio muy atractivo.

El ordenador VG 8235

El primero de la nueva gama de modelos MSX-2, el VG 8235, incorpora una unidad de disco de 3,5" con una capacidad de 360 Kb, 256 Kb RAM, pantalla de 80 columnas y funciones realzadas de color y gráficos.

Interfaces incorporados para impresora, lecto-grabadora y unidad de disco adicional, salida de monitor y TV, conectores de entrada/salida para joysticks, ratón y tableta gráfica y 2 ranuras para cartuchos ROM/RAM.

Paquete de software para la oficina en casa

El software de Philips "Home Office", que acompaña al MSX-2, está separado en 2 paquetes:

MSX Editor: Un paquete de procesador de textos profesional para preparación de alta calidad de todo tipo de documentación, como correspondencia e informes.

MSX Filer: Un programa de base de datos para un rápido y eficiente almacenaje y recuperación de información, tal como nombres, direcciones y números de teléfono.

MSX Editor y MSX Filer pueden usarse en combinación para aplicaciones de correo personalizado o similares.

Además, Philips ofrece un tercer programa con el MSX-2 llamado MSX Designer.

Es un sofisticado paquete de gráficos con Menú-directorio que permite al usuario mezclar color o diseños monocromos con textos, usando el teclado, ratón o tableta de gráficos.

Ascendencia total de compatibilidad MSX

Philips MSX garantiza la total compatibilidad en ascenso, permitiendo que todos los periféricos MSX y software se utilicen con el Philips MSX-2.

Philips MSX-2: El sistema completo para las aplicaciones de la oficina en casa.



Servicio de información al simpatizante y usuario.
Tels. (91) 413 21 61 - 413 22 46

PHILIPS

```
960 NEXT J
970 SPRITE$(I)=C$
980 NEXT I
990 REM PAISAJE
1000 DATA 0,0,0,0,0,0,0,0,0,0,0
     ,0,0,0,0,0,0,0,0,0,0,0,7,1
     5,31,15,3,0,0,0,0,0
1010 DATA 0,0,0,0,0,14,255,223,
     239,255,240,0,0,0,0,0,0,0,
     0,0,0,0,252,255,255,255,0,
     0,0,0,0,0
1020 DATA 0,0,0,0,0,0,0,255,255
     ,254,48,0,0,0,0,0,0,0,0,0,
     0,0,0,0,45,0,0,0,0,0,0,0
1030 DATA 0,0,2,66,34,16,3,7,23
     9,15,7,3,16,34,66,130,0,0,
     0,8,16,32,0,128,192,220,12
     8,0,32,16,8,4
1040 DATA 0,0,0,0,0,0,1,2,2,0,0
     ,0,0,0,7,15,0,0,0,0,96,128
     ,128,0,0,0,0,0,0,128,193,2
     27
1050 DATA 0,0,0,0,0,0,0,0,0,0,0
     ,0,48,248,255,255,0,0,0,0,
     0,0,0,0,0,1,3,7,15,31,63,2
     55
1060 DATA 0,0,60,63,60,32,96,96
     ,240,248,188,190,175,175,1
     75,47,1,0,4,8,0,32,0,0,0,0
     ,0,0,0,0,128,192
1070 DATA 2,0,2,2,0,2,0,0,0,0,0
     ,0,0,0,0,0,0,0,0,0,0,0,0,0
     ,0,0,0,0,0,0,7,15
1080 DATA 255,0,0,24,6,0,0,0,0,
     0,0,0,0,0,0,0,255,0,0,0,0,
     0,0,0,0,1,3,0,15,7,3,1
1090 DATA 255,1,3,7,15,31,63,12
     7,255,255,255,0,255,255,22
     6,238,255,255,255,255,255,
     255,255,255,255,255,255,0,
     255,255,139,187
1100 DATA 47,47,47,47,47,191,19
     1,191,191,224,224,32,255,2
     55,31,191,255,224,240,248,
     248,252,254,255,255,0,0,0,
     255,255,255,255
1110 DATA 255,0,24,6,0,0,0,0,6,
     1,0,0,255,255,255,255,255,
     0,0,0,0,0,0,0,0,128,0,0,22
     4,192,128,0
1120 DATA 255,255,255,255,255,2
     55,255,255,255,255,255,255
     ,255,255,255,255,255,255,2
     55,255,255,255,255,255,255
     ,255,255,255,255,255,255,2
     55
1130 DATA 230,110,63,31,15,7,0,
     0,0,0,0,0,0,0,0,155,187,
     255,255,255,255,0,0,0,0,0,
     0,0,0,0,0
1140 DATA 191,31,255,255,255,25
     5,0,0,0,0,0,0,0,0,0,255,
     255,255,255,255,255,0,0,0,
     0,0,0,0,0,0,0
1150 DATA 254,252,248,240,224,1
     92,0,0,0,3,0,0,0,0,0,0,0
     ,0,0,0,0,0,0,0,192,0,0,0
     ,0,0
1160 REM MSX
1170 DATA 0,0,0,0,0,0,0,0,0,0,0
     ,0,0,0,15,15,0,0,0,0,0,0,0
     ,0,0,0,0,0,7,8,250,250
1180 DATA 0,0,0,0,0,0,1,0,2,2,2
     ,2,242,8,175,175,0,0,5,32,
     64,128,60,66,66,66,90,90,6
     0,0,252,248
1190 DATA 0,0,160,4,2,1,60,66,6
     6,66,90,90,60,0,63,31,0,0,
     0,0,0,0,128,0,64,64,64,64,
     79,16,245,245
1200 DATA 0,0,0,0,0,0,0,0,0,0,0
     ,0,224,16,95,95,0,0,0,0,0,
     0,0,0,0,0,0,0,0,240,240
1210 DATA 15,14,14,14,14,14,14,
     14,14,14,14,14,14,14,14,14
     ,250,6,2,1,0,0,0,0,0,0,0,2
     4,28,30,27,25
1220 DATA 175,168,168,80,0,0,0,
     0,0,0,0,3,7,15,27,179,240,
     16,16,16,16,16,16,8,4,3,0,
     15,31,48,48,48
1230 DATA 15,8,8,8,8,8,8,16,32,
     192,0,255,255,0,0,0,245,21
     ,21,10,0,0,0,0,0,0,128,1
     92,96,48,25
1240 DATA 95,96,64,128,0,0,0,0,
     0,0,0,24,48,96,192,128,240
     ,112,112,112,112,112,112,1
     12,112,112,112,112,112,112
     ,112,112
```

```
1250 DATA 14,14,14,14,14,14,14,
     14,14,14,14,14,14,14,14,14
     ,24,24,24,24,24,24,24,24,0
     ,0,0,0,0,0,0,0
1260 DATA 227,67,3,3,3,3,3,3,0,
     0,0,0,0,0,0,0,48,31,15,0,0
     ,0,255,255,0,0,0,0,0,0,0,0
1270 DATA 0,240,248,12,12,12,24
     8,240,0,0,0,0,0,0,0,0,15,6
     ,15,25,48,96,192,128,0,0,0
     ,0,0,0,0,0
1280 DATA 0,0,0,128,192,96,48,2
     4,0,0,0,0,0,0,0,0,112,112,
     112,112,112,112,112,112,11
     2,112,112,112,112,112,112,
     112
1290 DATA 255,255,255,255,255,2
     55,255,255,255,255,255,255
     ,255,255,255,255,255,255,2
     55,255,255,255,255,255,255
     ,255,255,255,255,255,255,2
     55
1300 DATA 255,255,255,0,0,0,0,0
     ,0,0,0,0,0,0,0,0,255,255,2
     55,0,0,0,0,0,0,0,0,0,0,0,0
     ,0
1310 DATA 255,255,255,0,0,0,0,0
     ,0,0,0,0,0,0,0,0,255,255,2
     55,0,0,0,0,0,0,0,0,0,0,0,0
     ,0
1320 DATA 255,255,255,0,0,0,0,0
     ,0,0,0,0,0,0,0,0,240,240,2
     40,0,0,0,0,0,0,0,0,0,0,0,0
     ,0
1330 LINE(137,0)-(255,191),9,BF
1340 LINE(0,129)-(255,191),9,BF
1350 PRESET(150,40):PRINT#1,"U.
     ..Arriba"
1360 PRESET(150,50):PRINT#1,"N.
     ..Abajo"
1370 PRESET(150,60):PRINT#1,"H.
     ..Izquierda"
1380 PRESET(150,70):PRINT#1,"J.
     ..Derecha"
1390 PRESET(150,80):PRINT#1,"V.
     ..Modelo"
```

```
1400 LINE(116,150)-(164,158),9,
     BF
1410 PRESET(20,150):PRINT#1,"Mo
     vimientos ";USING"#####";M
     O
1420 FOR I=1 TO 4:FOR J=1 TO 4:
     MA(J,I)=T:T=T+1:NEXT J:NEX
     T I
1430 FOR I=1 TO 40:REM DESORDEN
     A
1440 R1=FIX(RND(-TIME)*4)+1
1450 R2=FIX(RND(-TIME)*4)+1
1460 R3=FIX(RND(-TIME)*4)+1
1470 R4=FIX(RND(-TIME)*4)+1
1480 D=MA(R1,R2):MA(R1,R2)=MA(R
     3,R4):MA(R3,R4)=D
1490 NEXT I
1500 FOR I=1 TO 4
1510 FOR J=1 TO 4
1520 IF MA(J,I)=12 THEN 1550
1530 PUT SPRITE MA(J,I),(J*32-3
     2+10,I*32-32),15
1540 GOTO 1560
1550 PUT SPRITE MA(J,I),(J*32-3
     2+10,I*32-32),8
1560 NEXT J
1570 NEXT I
1580 LINE(15,50)-(119,58),1,BF
1590 RETURN
1600 '
1610 '
1620 REM VER MODELO
1630 T=0
1640 FOR I=1 TO 4
1650 FOR J=1 TO 4
1660 PUT SPRITE T,(J*32-32+10,I
     *32-32),15:T=T+1
1670 NEXT J
1680 NEXT I
1690 PUT SPRITE 12,(10,32*4-32)
     ,8
1700 A$=INKEY$:IF A$="" THEN 17
     00
1710 IF A$="V" OR A$="v" THEN 1
     730
1720 GOTO 1700
1730 GOSUB 1500
1740 RETURN
1750 '
1760 '
1770 REM BUSCA CUADRADO ROJO
1780 FOR I=1 TO 4
1790 FOR J=1 TO 4
```

MICROBYTE PRESENTA... MSX

SENSACIONALES PROGRAMAS EN CARTUCHO Y CASSETTE

FLIGHT PATH 737.



Colócate a los mandos de un jet comercial. Disponemos de control total sobre los mandos del avión, y puedes escoger entre 6 niveles de dificultad.

P.V.P.: CART. 3.490 pts.
CASS. 1.900 pts. 32K.

FRUITY FRANK



Tu jardín ha sido invádido por monstruos de fruta madura. La única forma de combatirlos es lanzarles fruta fresca del jardín.

P.V.P.: CASS. 1.900 pts. 32K.

SPARTAN X



Son muchos los peligros que te acechan. Ten los reflejos bien despiertos, pon tus fuerzas en estado de alerta, y a luchar.

P.V.P.: CASS. 1.900 pts. 32K.

CHUCKIE EGG



Debes recoger los huevos antes de que nazcan los pollitos y se cóman el maíz. Pero ojo con el Pato Loco.

P.V.P.: CASS. 1.900 pts. 32K.

NIGHT FLIGHT



Con tu pequeño avión, debes ir dando luz a la noche, hasta que el cielo esté de nuevo azul. Date prisa en realizar tu misión, de lo contrario...

P.V.P.: CART. 2.900 pts.
CASS. 1.900 pts. 16K

STAR AVENGER



Imagina el juego de batalla más rápido que jamás hayas visto. Piensa además, en los más excitantes gráficos y sus 5 niveles de dificultad. Todo ello es Star Avenger.

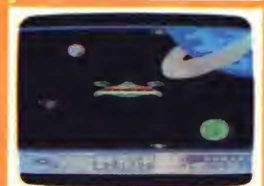
P.V.P.: CASS. 1.900 pts. 32K.

GYRO ADVENTURE



Ponte a los mandos de tu helicóptero y combate a los enemigos que se enfrentan a tí. Podrás mover el helicóptero en todas direcciones, mantenerlo en el aire y disparar. **P.V.P.: CART. 2.900 pts. CASS. 1.900 pts. 16K.**

SUPER CROSS FORCE



Sólo queda una esperanza para la supervivencia ante el ataque de los malvados Morpul. Tú podrás atacarles, con tus naves dispuestas en paralelo o en diagonal.

P.V.P.: CART. 2.900 pts. 16K.

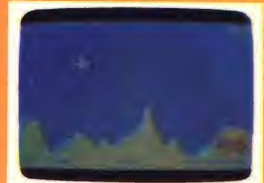
JUMP LAND



Tu mayor obsesión han sido siempre los pasteles, y por ello, te has visto envuelto en situaciones complicadas que has salvado gracias a tus reflejos.

P.V.P. CASS. 1.900 pts. 16K.

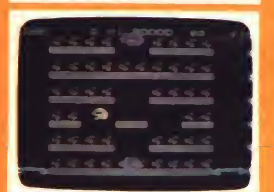
ROGER RUBBISH



Los perversos contaminadores de planetas están llenando nuestra galaxia de residuos nucleares. Roger Rubbish es el más famoso recogedor de basuras espaciales.

P.V.P.: CART. 2.900 pts. 16K.

FRUIT PANIC



Un día, Walky, para divertirse se fué al país de los gatos. ¿Cuánta fruta podrá comerse Walky?

P.V.P.: CASS. 2.000 pts. 16K

DIZZY BALLOON



En este mundo hay seres voladores y atacan cuerpo a cuerpo. Si los haces explotar, se irá abriendo el cielo y tendrás la oportunidad de escapar.

P.V.P.: CASS. 2.000 pts. 32K.

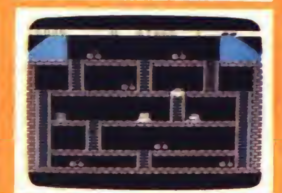
CASTLE COMBAT



El castillo galáctico, ha caído bajo la dominación de los Tyrones. Tu nave STAR DUSTER, está preparada para el combate. ¿Te atreves?

P.V.P.: CART. 2.900 pts. 16K.

NICK NEAKER



Cuando estás dormido, muchas cosas suceden a tu alrededor. Algunos objetos de tu casa toman vida, como en el caso de la zapatilla NICK.

P.V.P.: CART. 2.900 pts.
CASS. 1.900 pts. 16K.

CHAMP



Champ es un completo Ensamblador/Monitor para tu MSX. Champ te permite escribir y trazar programas en código máquina con el mínimo esfuerzo.

P.V.P.: CART. 3.890 pts.
CASS. 2.400 pts. 32K.

KARATE



Has conseguido entrar en la cueva de los piratas y ahora comienzan tus problemas. Los murciélagos gigantes, moradores de estas cuevas pueden chuparte la sangre. Cuando te encuentres con los piratas, deberás enfrentarte a ellos con tu depurado estilo de Kárate. **P.V.P.: CART. 3.490 pts. CASS. 1.900 pts. 32K.**

GRAND NATIONAL



Si te gustan las carreras de caballos, no te quedes como un espectador, participa. Ahora puedes correr con tu caballo, en la más prestigiosa carrera del mundo, el GRAND NATIONAL.

P.V.P.: CASS. 2.000 pts. 32K.

ENVIENOS A **MICROBYTE**

P.º Castellana, 179, 1.º - 28046 Madrid

Nombre
Apellidos
Dirección
Población
D.P. Teléfono

ENVIOS GRATIS

JUEGO	Cart	Cass.	Precio	TOTAL

PRECIO TOTAL PESETAS

Incluyo talón nominativo ☐
Contra-Reembolso ☐

Pedidos por teléfono **91 - 442 54 33 / 44**

```
1800 IF MA(J,I)=12 THEN X=J:Y=I
1810 NEXT J
1820 NEXT I
1830 RETURN
1840 '
1850 '
1860 REM COMPLETO ?
1870 T=0
1880 FOR I=1 TO 4
1890 FOR J=1 TO 4
1900 IF MA(J,I)<>T THEN RETURN
1910 T=T+1
1920 NEXT J
1930 NEXT I
1940 PRESET(20,170):PRINT#1,"Qu
     ieres jugar de nuevo? (s/n
     )"
1950 A$=INKEY$:IF A$="" THEN 19
     50
1960 IF A$="S" OR A$="s" THEN S
     CREEN 0:CLEAR:GOTO 10
1970 IF A$="N" OR A$="n" THEN E
     ND
1980 GOTO 1950
1990 '
2000 '
2010 REM CUENTA MOVIMIENTOS
2020 MO=MO+1
2030 LINE(116,150)-(164,158),9,
     BF
2040 PRESET(20,150):PRINT#1,"Mo
     vimientos ";USING"####";M
     O
2050 BEEP
2060 RETURN
2070 '
2080 '
2090 REM PRESENTAR OPCION
2100 SCREEN 2,2
2110 S1=30:S2=15
2120 FOR W=1 TO 2
2130 FOR I1=0 TO 15
2140 C$=""
2150 FOR J1=1 TO 32
2160 READ P
2170 C$=C$+CHR$(P)
2180 NEXT J1
2190 SPRITE$(I1+M)=C$
2200 NEXT I1:T=0
2210 LINE(0,0)-(29,191),9,BF
2220 LINE(95,0)-(155,191),9,BF
2230 LINE(219,0)-(255,191),9,BF
2240 LINE(0,0)-(255,14),9,BF
2250 LINE(0,80)-(255,109),9,BF
2260 LINE(0,175)-(255,191),9,BF
2270 LINE(156,15)-(220,79),9,BF
2280 LINE(30,110)-(94,174),9,BF
2290 FOR I1=1 TO 4
2300 FOR J1=1 TO 4
2310 IF T<>12 THEN PUT SPRITE T
     +M,(J1*16-16+S1,I1*16-16+S
     2),15 ELSE PUT SPRITE T+M,
     (J1*16-16+S1,I1*16-16+S2),
     8
2320 T=T+1
2330 NEXT J1,I1:S2=110:S1=156:M
     =16
2340 NEXT W
2350 OPEN "GRP:" AS 1
2360 PRESET(70,170):PRINT#1,"Op
     cion ?"
2370 PRESET(54,100):PRINT#1,"1"
2380 PRESET(180,90):PRINT#1,"2"
2390 A$=INKEY$:IF A$="" THEN 23
     90
2400 IF A$="1" THEN CLEAR:SCREE
     N 0:CLS:SCREEN 2,3:RESTORE
      1000:GOTO 110
2410 IF A$="2" THEN CLEAR:SCREE
     N 0:CLS:SCREEN 2,3:RESTORE
      1170:GOTO 110
2420 GOTO 2390
2430 '
2440 '
2450 REM INSTRUCCIONES
2460 CLS
2470 LOCATE 0,1:PRINT"El juego
     consiste en lograr que el
       conjunto de piezas que f
     orman el gra-fico queden c
     olocadas correctamente."
2480 LOCATE 0,5:PRINT"Para ello
      deberas ir moviendo algun
     a de las piezas que se enc
     uentran alre-dedor de la c
     asilla vacia (en color  ro
     jo) hacia esta."
2490 LOCATE 0,15:PRINT"    Puls
     a una tecla para empezar"
2500 A$=INKEY$:IF A$="" THEN 25
```

SERVICIO DE EJEMPLARES ATRASADOS

ESTOS SON LOS EJEMPLARES DE MSX MAGAZINE APARECIDOS EN EL MERCADO CON UN RESUMEN DE SU CONTENIDO



Núm. 1
¿Qué es el MSX? Su BASIC. periféricos, programas, software.



Núm. 2
Generacion de sonido. MSX-DOS. el ordenador por dentro. programas. noticias.



Núm. 3
Los joysticks, 256 caracteres programables, Z80 corazón de león, compro/vendo/cambio.



Núm. 4
Las comunicaciones entre ordenadores, la jerga informática, trucos, rincón del lector.



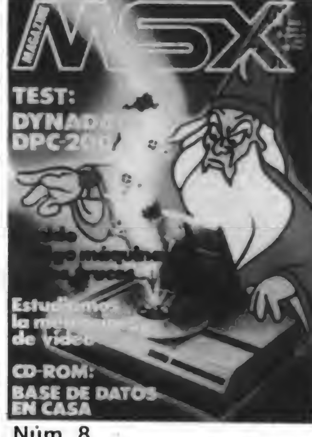
Núm. 5
Comandos de entrada/salida, el BASIC MSX comparado con Spectrum y Commodore 64. Código Máquina.



Núm. 6
Los 8 magnificos (test gigante), el bus de expansión, los misterios de la grabación, programas.



Núm. 7
Analizamos el Generador de Sonido. Aplicaciones matematicas con el ordenador. La memoria de video. Trucos, noticias.



Núm. 8
Compact Disc, el periférico del futuro. Test: Dynadata DPC-200. Continuamos con la memoria de video. Libros. software, programas, trucos.



Núm. 9
Caracteristicas tecnicas del Compact Disc. Tratamiento de datos. Test: Quick Disk. Trucos, libros, noticias, programas.

PARA HACER SU PEDIDO, RELLENE ESTE CUPON, HOY MISMO Y ENVIELO A MSX MAGAZINE BRAVO MURILLO, 377. Tel. 733 96 62 - 28020 MADRID

Ruego me envien los siguientes numeros atrasados ....

al precio de **250** ptas cada uno. Cuyo importe abonare:

☐ POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO

☐ AMERICAN EXPRESS ☐ VISA ☐ INTERBANK

Numero de mi tarjeta ....

Fecha de caducidad ....

NOMBRE ....

DIRECCION ....

POBLACION .... C.P. ....

PROVINCIA ....

Código Máquina

En esta ocasión trataremos tres grupos de instrucciones muy importantes: las instrucciones aritméticas y de control de la CPU, las instrucciones BIT, SET y RES, y el grupo de saltos.

La *CPU* puede realizar algunas operaciones aritméticas y lógicas muy concretas que permiten el control del estado de la *CPU* y del grupo general de instrucciones aritméticas de 8 *bits*. El usuario recibe información del estado de la *CPU* a través del registro de *flag* (F). Este registro está formado por 8 *bits*, de los cuales seis proporcionan información, y dos no se utilizan. Estos *bits* son los siguientes:

S Z X H X P/V N C

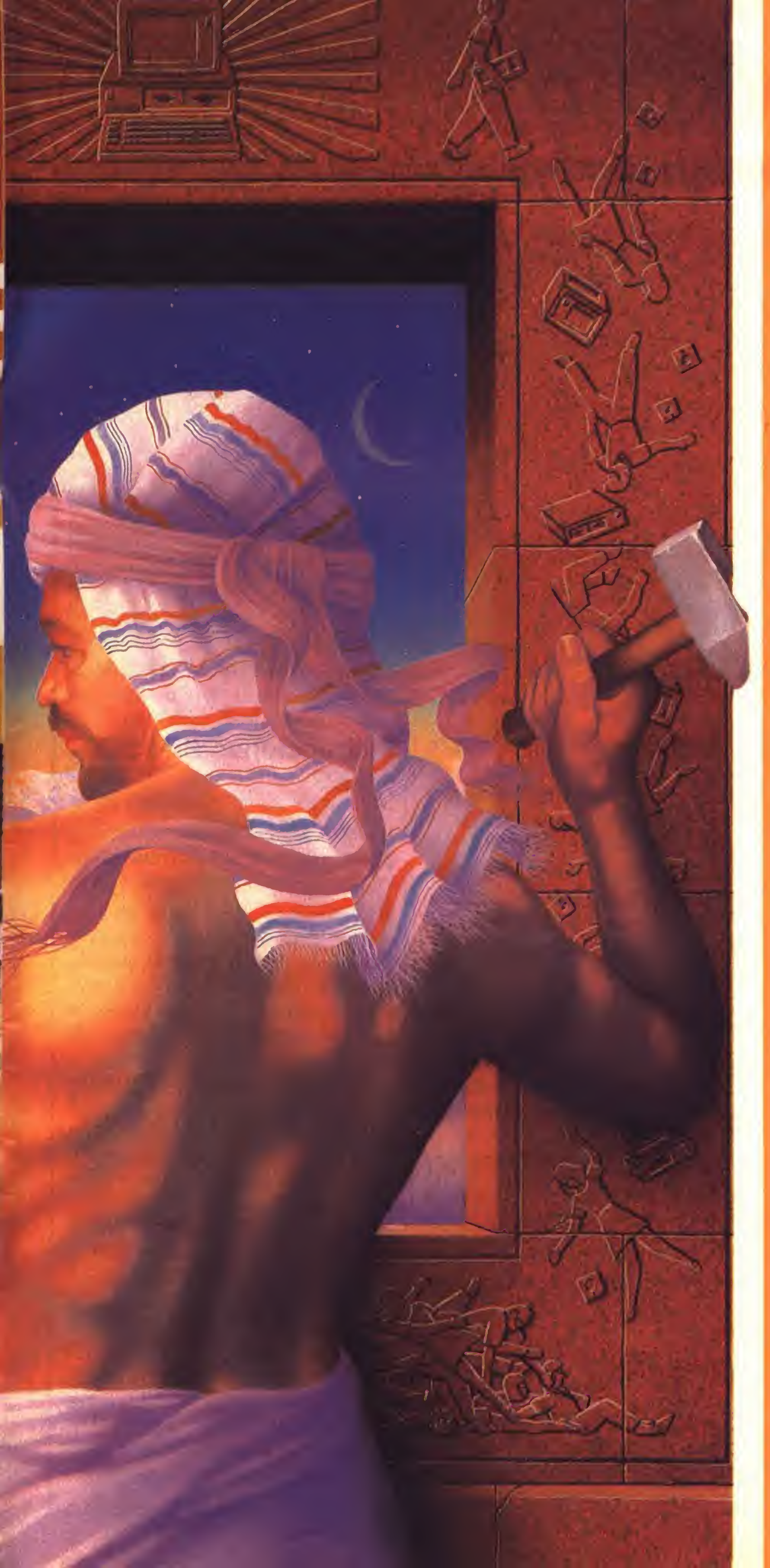
Los marcados como X son los que no se utilizan. El *flag* S (signo), indica si el *bit* más significativo del resultado de una operación es 0 o 1. Si es 1, tal resultado se considera negativo en aritmética en complemento a dos de 8 *bits*, y si es cero se considera positivo. El *flag* Z (cero) se activa (se pone a 1) cuando el acumulador es cero COMO RESULTADO DE UNA OPERACION QUE AFECTE A ESTE FLAG. Y marcamos esto porque es importante tener en cuenta que no todas las operaciones afectan a todos los *flags*.

El *frag H* (Half — Carry = Medio Acarreo) funciona de forma semejante al *flag* de acarreo (explicado más adelante), pero sólo con la aritmética BCD y referido al arrastre del cuarto al quinto *bit*.

El flag P/V puede tener dos significados, dependiendo de qué operación se trate: Paridad (P) o Sobrepasamiento (V). La paridad se refiere el número de unos que contiene un byte. Así, si el número de unos es par, el indicador P será 1, y si es impar, cero. El *flag* de sobrepasamiento se pone a 1 si el resultado de una operación aritmética necesita un noveno *bit* (en el caso de operaciones de 8 *bits*) o un decimoséptimo *bit* (en el caso de operaciones de 16 *bit*).

El *flag* N indica si la última operación aritmética realizada fue una recta, poniéndose a 1 si fue así. El *flag* C (Carry = Acarreo; algunas veces puede aparecer la abreviatura CY en lugar de C) indica si al realizar una operación aritmética, existió un acarreo en el último *bit*





(octavo o decimoséptimo, según el caso). Si existió tal acarreo, el indicador se pone a uno.

La importancia de los *flags* estriba en que existen instrucciones que permiten verificar el estado de cuatro de ellos (Z, P, N y C) y obrar en consecuencia: se trata de las instrucciones de salto condicional, que tratamos un poco más adelante, y de las llamadas a subrutina condicionales, que trataremos en un próximo capítulo.

Comencemos ya, sin más dilación, con las instrucciones de control de la *CPU:*

DAA: Estas siglas significan Ajuste de Aritmética Decimal. Se utilizan para realizar cálculos en aritmética *BCD* (Binario Codificado en Decimal), que permite utilizar un *byte* para representar dos dígitos decimales, separando el *byte* en dos grupos de 4 *byts,* de modo que cada grupo de 4 *bits* (nibble) representa un dígito decimal, según el siguiente código:

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	sin significado
1011	sin significado
1100	sin significado
1101	sin significado
1111	sin significado

Así, por ejemplo, el número 10010011, que en binario representa 147, representa en *BCD* 93 (1001 = 9, 001 = 3). Esta instrucción actúa directamente sobre el acumulador, convirtiendo su contenido a *BCD* aunque la instrucción anterior no sea aritmética, y

código máquina

afecta a los siguentes flags: C, Z, P, S y H.

Veamos un ejemplo de su uso: supongamos que queremos sumar los números decimales 27 y 35, utilizando aritmética decimal. El 27 se representa como 00100111, o lo que es lo mismo 027H, y el 35 por 00110101 (035H). Para sumarlos deberiamos hacer lo siguiente:

```
LD  A,027H
LD  B,035H
ADD B
DAA
```

Veamos lo que ocurre: una vez cargados los datos en los registros A y B, los sumandos con la instrucción *ADD B* (ver capítulo anterior). Como la *CPU* trabaja en binario, hemos efectuado la siguiente suma:

00100111	027H = 39 decimal.
+ 00110101	035H = 53 decimal.
01011100	05H = 92 decimal.

Si interpretamos el resultado como *BCD*, obtenemos un 5 y un nibble sin sentido. La instrucción *DDA* corrige esto, y convierte el contenido del acumulador (05H) en 062H, que es el resultado correcto en *BCD*.

CPL: Invierte el contenido del acumulador, *bit* a *bit*. Asi si tenemos en el acumulador 01010011, después de utilizar la instrucción *CPL* obtenemos en el acumulador 10101100. Esta instrucción pone a uno de los *flags* N y H.

NEG: Realiza la instrucción *CPL* y luego le suma uno al acumulador, obteniendo asi el "complemento a dos" del acumulador. Este complemento a dos es una forma de representar números que considera el octavo o decimosexto *bit* como el signo (i=menos, 0= más). Además, los números negativos se representan de este modo: **(Ver figura 1)**

Número:	8 *bits*	16 *bits*
− 1	11111111	1111111111111111
− 2	11111110	1111111111111110
− 3	11111101	1111111111111101
− 4	11111100	1111111111111100
− 5	11111011	1111111111111011
− 6	11111010	1111111111111010
"	"	"
"	"	"
"	"	"
−128	10000000	1111111110000000
"	"	"
"	"	"
"	"	"
−32768	--------------	1000000000000000

Figura 1

Esta instrucción altera el *flag* V.

CCF: (*Complement Carry Flag* = Complementa la bandera de acarreo). Esta instrucción invierte el *bit* 0 del registro de estado de la *CPU* (el marcado como C).

SCF: (Set Carry Flag). Fuerza el *bit* de acarreo (C) a 1.

NOP: No operación. Esta instrucción no hace nada más que ocupar un *byte* y utilizar cuatro ciclos de reloj.

HALT: Detiene la *CPU* hasta que se produzca una interrupción.

DI: Desautoriza las interrupciones enmascarables.

EI: Autoriza las interrupciones enmascarables.

IM 0: Fija el modo de interrupción 0.

IM 1: Fija el modo de interrupción 1.

IM 2: Fija el modo de interrupción 2.

Instrucciones BIT, SET y RES

Estas instrucciones permiten poner a 0, poner a 1 y probar el valor de cualquiera de los 8 *bits* de un registro independientemente:

BIT b,r.—b puede ser cualquier *bit* (0 a 7), y r puede ser B, C, D, E, H, L o A. El efecto de esta instrucción es activar el *flag* de cero (Z) si el *bit* es cero, y desactivarlo si es uno.

BIT b,(HL).—b igual que en la anteior. En este caso se prueba un *bit* del *byte* contenido en la dirección de memoria a la que apunta *HL* (ver capítulo anterior sobre el concepto de «puntero»).

BIT b,(IX+d).—b igual que en las anteriores. d es el desplazamiento que se suma al registro índice IX. Recordamos a los lectores que el registro IX no resulta afectado. Además, NINGUNA INSTRUCCION *BIT* AFECTA AL *BYTE* PROBADO. Así, si el registro B contiene 01001100 y efectuamos BIT 4,B, el flag Z se pondrá a 1, ya que el *bit* cuatro del registro B es cero (os re-

cordamos que se cuenta de derecha a izquierda y de 0 a 7), pero B sigue conteniendo 01001100.

BIT b, (IY+d). igual que la anteior, pero con el registro índice *IY.*

SET b,r.–b y r igual que en la primera instrucción *BIT.* En este caso, el efecto es fijar a 1 el *bit* indicado.

SET b, (HL). la alteración se produce en el *byte* al que apunta *HL.*

SET b, (IX+d) igual que la anterior, pero utilizando el registro índice *IX.*

SET b, (IY+d). igual que la anterior, pero utilizando el registro índice *IY.*

RES b,r.–b y *r* igual que en los casos anteriores. El efecto producido es fijar a cero el *bit* correspondiente.

RES b, (HL). la alteración se produce en el *byte* al que apunta *HL.*

RES b, (IX+d) RES b, (IY+d). igual que en los casos anteriores.

Dentro del grupo de instrucciones de salto podemos distinguir dos tipos: aquéllas en las que, tras el código de operación, se especifica la dirección a la cual se salta; y aquéllas en las que se especifica un número de 8 *bits* que hay que sumar a la dirección actual del Contador de Programa *(PC).* Veremos dentro de estas últimas un caso especial que combina dos instrucciones en una.

JP nn.–nn representa a un número de dieciséis *bits.* El efecto de la instrucción es saltar a la dirección especificada por el número *nn.* Hay que significar que la dirección de salto está escrita de forma que primero encontramos el *byte* bajo y luego el alto. Por ejemplo, si encontramos los códigos C3 1A B0, C3 es el código de instrucción de *JP nn,* y la dirección de salto no es 1AB0, sino B01A.

JP cc,nn. igual que la anterior, pero la ejecución de la instrucción depende de que se verifique o no una condición en el registro de estado de la *CPU.cc* puede ser:

NZ. se efectúa el salto si el *flag* Z es cero.

Z. se efectúa el salto si el *flag* Z es uno.

NC. se efectúa el salto si el *flag* C es cero.

C. se efectúa el salto si el *flag* C es uno.

PO. se efectúa el salto si el *flag* P es uno.

PE. se efectúa el salto si el *flag* P es cero.

P. se efectúa el salto si el *flag* S es cero.

M. se efectúa el salto si el *flag* S es uno.

JR d.–d representa el desplazamiento que hay que sumar a la dirección de la instrucción siguiente a la de salto para calcular la dirección efectiva de salto. Dicho desplazamiento es un método de 8 *bits* con signo en complemento a dos.

Aclaremos esto con un ejemplo: ***(Ver figura 2)***

Al llegar a la instrucción de salto (2204) la *CPU* coge el código de operación 18, incrementa el contador de programa, que ahora marca 2005, y analiza el código. Como *JR* necesita un dato, coge el dato *FB,* incrementa el contador de programa (2006) y ejecuta la instrucción. Para ello resta el dato del contador de programa, esto es, 2006 – 5 = 2001, y pasa el resultado al contador de programa, con lo cual está efectuado el salto.

JR cc,d. igual que *JR,* pero con la condición *cc* (como en el caso de *JP cc,nn*). En esta ocasión *cc* sólo puede ser *NZ, Z, NC, C.*

JP (HL) – JP (IX) – JP (IY). Estas tres instrucciones toman como dirección de salto el contenido del respectivo registro doble. Así, si *HL* contiene 207F, *JP (HL)* equivale a *JP* 207F. Ninguna instrucción de salto afecta a ningún *flag.*

Antes de terminar, vamos a explicar las interrupciones en el *Z80.* Como hemos visto anteriormente, la *CPU* puede funcionar en tres modos de interrupción: 0, 1 y 2. Veamos cada uno de ellos.

En el modo 0, cuando la *CPU* recibe una señal de interrupción busca en el bus de datos un código de *RST.* Las instrucciones *RST* son como las llamadas a subrutinas *(CALL),* con la particularidad de que no necesitan dirección de llamada, ya que se supone que el *byte* alto es 00, y existen 8 instrucciones, cada una de las cuales lleva implícito el byte bajo. Son las siguientes:

Dirección (Hex.)	Dato (Hex.)	Instrucción:
2000	78	LD A,B
2001	C6	ADD FFH
2002	FF	
2003	B1	OR C
2004	18	JR –5
2005	FB	
2006	00	NOP

Figura 2

código máquina

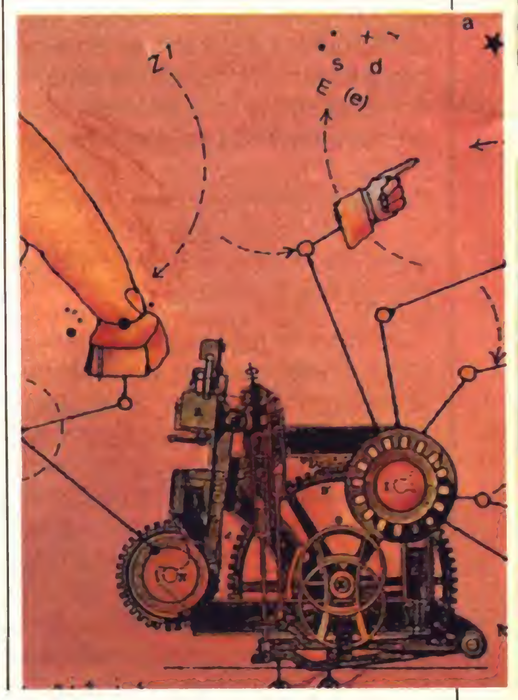
RST 0-equivale a *CALL* 0000H
RST 8-equivale a *CALL* 0008H
RST 10-equivale a *CALL* 0010H
RST 18-equivale a *CALL* 0018H
RST 20-equivale a *CALL* 0020H
RST 28-equivale a *CALL* 0028H
RST 30-equivale a *CALL* 0030H
RST 38-equivale a *CALL* 0038H

Este código debe ser enviado por el periférico que provocó la interrupción. De este modo, se pueden controlar hasta 8 periféricos, asignando a cada uno a una rutina de interrupción en una de las ocho direcciones de *RST*.

En el modo 1 la señal de interrupción genera siempre la misma instrucción *RST: RST 38*. Este es el modo en que funcionan los MSX y muchos otros ordenadores domésticos (AMSTRAD, SPECTRUM, etc.). Normalmente, en esta dirección hay una instrucción de salto a la rutina de lectura de teclado y otros procesos «automáticos» del ordenador.

El modo 2 es algo más completo. En primer lugar, el programador debe preparar en memoria una tabla de direcciones de dieciséis *bits*. Estas direcciones apuntan a las rutinas de interrupción que manejan cada periférico. Cuando se produce la interrupción, el periférico envía el *byte* bajo de la dirección de la tabla de direcciones en la cual se encuentra la dirección de salto, y la *CPU* toma el *byte* alto del registro I del *Z80*. Este registro puede ser cargado con la instrucción *LD I,A*, y ser leído con la instrucción *LD A,I*.

No os preocupéis por la falta de ejemplos. Primero vamos a explicar el juego de instrucciones, para que tengáis un texto al cual referiros cuando no recordéis cómo funciona una instrucción. Una vez completada esta parte «teórica», comenzaremos a aplicar estos conocimientos en el ordenador.



Idea de IDEALOGIC SA
Utiles con potencia



¡¡LA SOLUCION!!

IDEA BASE
- Acepta formato de etiquetas.
- Funciones de modificación, borrado y añadido de información en registros.
- Información almacenable en cinta o disco.
- Capacidad de 42 K en memoria.
- Menús conversacionales e interactivos. Máxima facilidad de uso.
- Ordenación de registros en función de uno o más campos.
- Presentación en cartucho.
- Gestiona disco y cinta.
- Compatible con IDEA TEXT y DIM-CALC

IDEA TEXT
- Editor de página entera con control total de márgenes, indentaciones, centrado, espacios, encabezados, pies de página, numerador automático, etc.
- Permite mover, reemplazar e insertar bloques de texto de una manera fácil.
- Función de visualización del texto final, permitiendo examinar cual será el resultado de la impresión.
- Capacidad de 42 K en memoria.
- Máxima facilidad de uso apareciendo constantemente información en la pantalla.
- Menús comprensivos en ventanas.
- Presentación en cartucho.
- Gestiona disco y cinta.
- Compatible con IDEA BASE y DIM-CAL.

ESTAMOS EN INFORMAT/86
STAND 911 NIVEL 9
PALACIO FERIAL

DIM-CALC.
Hoja de cálculo muy facil de utilizar, que permite desarrollar desde cálculos sencillos a otros complejos y sofisticados. Compatible con IDEA TEXT e IDEA BASE.

Solicite información a nuestro departamento de Marketing

IDEALOGIC® SA
Calle Valencia, 85 - 08029 BARCELONA - Télex 54554 DLGC
Teléfonos 253 86 93 - 253 89 09 - 253 90 45 - 253 74 00

Delegaciones:
Distribuidores Generales en: Almería, Badajoz, Barcelona, Bilbao, Cádiz, Córdoba, Gijón, Granada, La Coruña, Las Palmas, Lérida, Madrid, Málaga, Murcia, Oviedo, Palma Mallorca, Pamplona, Sevilla, Valencia, Valladolid, Vigo, Zaragoza, Argentina, Chile y México.

Rincón del lector

PROBLEMAS CON EL SPECTRAVIDEO

Cuando compré el ordenador, me decidí por un SVI-328, por razones económicas y por su versatilidd, así como por ser un ordenador estándar.

En principio, no me he decidido a comprar una unidad de discos, la cual me resulta imprescindible porque me habían hablado que estaba a punto de aparecer una unidad de expansión similar a la del SVI-328. Tampoco me he decidido a ampliar el ordenador en espera de esa caja de expansión que no llega, ¿de quién es la culpa? ¿Llegará un día? ¿O es que me han tomado el pelo?

Quisiera decirles que, ni muchísimo menos esta herramienta fue adquirida para jugar y cual ha sido mi desilusión, cuando veo que los programas, tanto didácticos como semiprofesionales, tardan en aparecer e incluso me atrevería a decir que ni siquiera aparecen juegos de una cierta calidad a pesar de estar avalada la estandarización por tan importantes marcas. ¿Qué está sucediendo? ¿Hay una mano oculta que está impidiendo el desarrollo del estándar? Pues veo con sorpresa que ya se va a comercializar la 2.ª generación cuando los que poseemos uno de la primera no hemos podido aprovecharnos de los beneficios que se repercutirían del estándar. Y mi última pregunta, ¿cuándo van a comentar programas didácticos y de orientación profesional en disco?

Antonio Ruiz Alba
Madrid

No creo que te hayan tomado el pelo. La unidad de expansión de Spectravídeo para el 728, siempre fue un misterio, por lo menos mientras lo llevaba Indescomp. Nuestro consejo es que llames a Spectravídeo España, nueva empresa que se va a dedicar a comercializar todos los Spectravídeo y plantearles el tema. Los planes de expansión, abarcan todo el ámbito del Spectravídeo que Indescomp abandonó totalmente. Fruto de esto, muchos usuarios descontentos han escrito y otros, los menos, se han dejado llevar por las circunstancias. Si te sirve de ayuda, dirígete a la siguiente dirección y plantéales tus dudas respecto al SVI'728;

SVI España
Avda. de la Constitución, 260
Torrejón de Ardoz
Madrid

Respecto al software, es poco lo que hay en disco, pero lo que hay es francamente bueno. Actualmente estamos trabajando con un tratamiento de textos (desde los incios de **MSX Magazine** hemos trabajado con la versión holandesa de éste, pero que en nuestro país lo comercializa Philips bajo la denominación de MS-Text) y una potente base de datos que comentaremos a fondo en sucesivos números.

PASAR PROGRAMAS DE SV-328 A MSX

Poseo el SV-328 MKII, y aunque la mayoría de los programas son compatibles con MSX, a veces me encuentro con algunos programas que no lo son, y ahí me atasco. También me encuentro con algunos mensajes de error pero estos los achaco a la configuración de mi ordenador.

Santiago Segura Merino
Granada

Efectivamente, el SV-328 es casi un MSX, ya que el lenguaje es el mismo con algunas excepciones. Las variaciones necesarias para alterar cualquier programa MSX las encontrarás en el número 8 del mes de diciembre de MSX Magazine. De cualquier manera, continuaremos publicando trucos e ideas para mejorar las posibilidades del SV-328 y, al menos, equipararlo a su hermano SV-728.

MSX

DYNADATA

NO LO DUDE

PRECIO INSUPERABLE

CONJUNTO:
ORDENADOR
MONITOR Y
CASSETTE
DYNADATA MSX

62.500

TARJETA
PROGRAMA.
EL NUEVO FORMATO DEL FUTURO.

MONITOR 12″
FOSFORO
VERDE

ADAPTADOR TARJETA

DATA CASSETTE

NUEVO

DYNADATA MSX
CON TECLADO EN
ESPAÑOL*

*Letras y signos iguales
al teclado del PC de IBM.

Ordenador DYNADATA MSX con teclado en español	46.900
DYNADATA MSX con cassette y monitor de color	95.500
DYNADATA MSX y unidad de diskette de 5 ¼″ de 360 Kbytes con monitor de fósforo verde	108.900
con monitor de color	141.000
TARJETA/PROGRAMA del juego LE MANS con adaptador, el cual sirve para cualquier tarjeta que usted adquiera. Precio especial con la compra del DYNADATA MSX	4.900

Con el DYNADATA MSX usted podrá:

- Divertirse con la amplia gama de juegos MSX.
- Aprender Informática y Basic con el curso autodidáctico y audiovisual.
- Llevar gestiones administrativas con los programas de proceso de textos, base de datos, contabilidad, stock, recibos, etc.
- Ayudar a sus hijos en sus estudios de 3.º a 8.º de EGB con los programas de Matemáticas, Lenguaje y Ciencias Naturales.
- Aprender idiomas tan necesarios como el inglés.
- Programar con los lenguajes: LOGO, PASCAL, FORTRAN y COBOL.

CURSO DE INFORMATICA Y BASIC

- Autodidáctico
- Audiovisual
- 12 cassettes
- 24 lecciones
- Evaluaciones periódicas
- Diploma Fin de Curso



DYNADATA MSX
Y CURSO
69.300

Especificaciones DYNADATA MSX: Procesador Z80A, 64 Kbyte RAM, 16 Kbyte VRAM, 32 Kbyte ROM, 24 líneas x 40 columnas, 256 x 192 pixels, 16 colores, MSX-BASIC, MSX-DOS.

DYNADATA

Y:

	PIONEER
SONY	YAMAHA
PHILIPS	MITSUBISHI
CANON	GOLDSTAR
SANYO	SAMSUNG
JVC	HITACHI
TOSHIBA	MATSUSHITA
SPECTRAVIDEO	CASIO

Se han decidido por MSX. Esto le permite compartir los programas y periféricos con todas estas reconocidas marcas.

Por todo, NO LO DUDE. Decídase por



Solicite información: Sor Angela de la Cruz, 24 - 28020 Madrid. Tels. (91) 279 21 85 - 279 28 01 - 270 01 93. Telex 44619 DYNA
Delegación Barcelona: Aribau, 61, entlo - 08011 Barcelona. Tels. (93) 254 73 04 - 254 73 03